Onno W. Purbo | STKIP Surya

JARINGAN MESH

Solusi Jitu Membangun
Jaringan Wireless Gotong Royong
Tanpa **Access Point**

Jaringan MESH

Solusi Jitu Membangun Jaringan Wireless Gotong Royong Tanpa Access Point

Jaringan MESH

Solusi Jitu Membangun Jaringan Wireless Gotong Royong Tanpa Access Point

Onno W. Purbo

STKIP Surya

Penerbit ANDI Yogyakarta

Jaringan MESH - Solusi Jitu Membangun Jaringan Wireless Gotong Royong Tanpa Access Point

Oleh: Onno W. Purbo & STKIP Surya

Editor : Fl. Sigit Suyantoro
Setting : Alek
Desain Cover : dan_dut
Korektor : Susy Oktaviani

Penerbit: C.V ANDI OFFSET (Penerbit ANDI)
Jl. Beo 38-40, Telp. (0274) 561881 (Hunting), Fax. (0274) 588282
Yogyakarta 55281

Percetakan: ANDI OFFSET
Jl. Beo 38-40, Telp. (0274) 561881 (Hunting), Fax. (0274) 588282
Yogyakarta 55281

Perpustakaan Nasional: Katalog dalam Terbitan (KDT)

Purbo, Onno W

Jaringan MESH – Solusi Jitu Membangun Jaringan Wireless Gotong Royong Tanpa Access Point/Onno W. Purbo & STKIP Surya;
– **Ed. I.** – Yogyakarta: ANDI,

22 21 20 19 18 17 16 15 14 13

x + 134 hlm.; 14 x 21 Cm.

10 9 8 7 6 5 4 3 2 1

ISBN: 978 – 979 – 29 – 3429 – 8

I. Judul

1. Wireless Networks

2. Surya, STKIP

DDC'21 : 384.545.5

KATA PENGANTAR

Saat kita menghadapi medan bergunung-gunung, listrik dan infrastruktur seperti tower langka, mau tidak mau kita harus berpikir keras mencari solusi supaya kita dapat membangun jaringan yang murah tanpa perlu investasi terlalu banyak.

Wireless memang memungkinkan kita membangun jaringan yang murah, tetapi wireless yang normal membutuhkan banyak tower untuk menempatkan Base Station/Access Point. Dengan menggunakan teknologi mesh maka, kita dapat meminimalkan investasi tower yang mahal. Dengan teknologi mesh kita dapat memaksimalkan mekanisme gotong royong antar laptop/PC, dll.

Teknologi Mesh memungkinkan kita membangun jaringan di daerah yang kurang baik, yang bergunung, yang tidak tersedia cukup uang untuk investasi.

Penulis mengucapkan terima kasih kepada teman-teman dari:

- FreiFunk Berlin, Elektra dan teman-teman yang membuka wawasan penulis tentang Mesh.
- Sebastian Buettrich & Tomas Krag dari Wireless.dk yang mengundang penulis ke Denmark untuk memperkenalkan Mesh Network.
- Pak Teguh Wijaya & teman-teman di UBNT Indonesia yang meminjamkan peralatan untuk dioprek.

- Pak Andy, Bu Kunti & teman-teman dari Buffalo Indonesia yang meminjamkan peralatan untuk dioprek.

Semoga buku ini dapat membuka wawasan pembaca dalam pengembangan jaringan yang murah di Indonesia.

Jakarta, September 2012

Onno W. Purbo

DAFTAR ISI

KATA PENGANTAR iii

DAFTAR ISI v

BAB 1 MESH NETWORK 1

Jaringan Wireless Mesh 2

Contoh Jaringan Mesh Di Dunia 3

Beberapa Skenario Jaringan Mesh 5

BAB 2 TEKNIK ROUTING UNTUK MESH 11

Optimized Link State Routing Protocol 12

Fitur yang spesifik di OLSR 14

Keuntungan 15

Kritik Terhadap OLSR 15

Message di OLSR 16

Pendekatan Lain 17

OLSR versi 2 18

Implementasi 18

BAB 3 INSTALASI OLSR DI KOMPUTER 21

Instalasi OLSR di Ubuntu 21

Compile dari Source Code di Ubuntu 22

Edit Konfigurasi OLSR 23

Cek konfigurasi OLSR 25
Edit Interface.. 25
Restart Networking.. 27
Verifikasi Jaringan 28
Cek Routing.. 29
OLSR di Windows.. 30

BAB 4 FREIFUNK FRIMWARE 33
Komunitas Mesh FreiFunk.. 33
FREIFUNK Firmware/OPENWRT.. 34
Fitur FreiFunk Firmware.. 34
Firmware PLUGIN.. 35
LAYANAN LOKAL di jaringan mesh.. 35
MENGADAPTASI FIRMWARE.. 35
Komunitas di Dunia 36
Linksys WRT54GL Upgrade FreiFunk Firmware......... 36
Linksys WRT54GL Konfigurasi FreiFunk.................... 38
Linksys WRT54GL FreiFunk Firmware Konfigurasi OLSR.. 42
Konfigurasi Linksys WRT54GL FreiFunk Firmware Wireless Setting 43
Konfigurasi Linksys WRT54GL FreiFunk Firmware LAN & WAN Setting.. 45
Linksys WRT54GL FreiFunk Firmware Update & Upload Software.. 47
Linksys WRT54GL FreiFunk Firmware Restart Device.. 50

Tampilan Linksys WRT54GL FreiFunk Firmware Setelah Upgrade Aplikasi 51

BAB 5 MEMBUAT SISTEM OPERASI OPENWRT 57

Download Firmware OpenWRT yang sudah Jadi 57

Prosedur Compile Firmware OpenWRT 59

Beberapa Tips Saat Compile Firmware OpenWRT 61

Menentukan Chipset 61

Build di multi-core CPU 61

Build di background 62

Build single Packages 62

Melihat build errors 62

Notifikasi Beep 63

Lokasi Image Firmware Hasil Compile 63

Membersihkan / Clean Up 63

Melihat Daftar Package 64

Contoh Proses Compile OpenWRT Firmware Untuk Buffalo WZR HP G450H 65

Setup Menuconfig 67

Contoh Proses Compile OpenWRT Firmware Untuk UBNT 73

BAB 6 INSTALASI DAN OPERASI OPENWRT PADA ROUTER DAN AP 85

Teknik Recovery UBNT 85

Teknik Recovery Buffalo 89

Setup Identitas Saat Pertama Kali Menggunakan OpenWRT 93

Instalasi & Operasi OLSR di UBNT Menggunakan Backfire OpenWRT.. 95
Konfigurasi OLSRD UBNT NanoStation2 dengan Firmware OpenWRT buatan Sendiri dan WebGUI. ... 106
Konfigurasi OLSRD di UBNT NanoStation2 Tanpa WebGUI 106
Instalasi Aplikasi Pendukung OLSRD di Buffalo WZRHPG450H Menggunakan OpenWRT.................. 110
Sedikit Catatan Tentang Firewall di OpenWRT 116

BAB 7 SAMBUNGAN KE INTERNET DAN JARINGAN LAIN..... 119
Sambungan menggunakan LAN ke Internet 119
Sambungan menggunakan LAN ke jaringan lain/LAN. 120
Sambungan menggunakan modem 3G..................... 120
Banyak Sambungan ke Internet................................ 125

BAB 8 PENUTUP 129

REFERENSI.. 131

MESH NETWORK

Mesh networking (*topology*) adalah salah satu jenis jaringan di mana setiap node di jaringan tidak hanya menerima atau mengirim data miliknya, tetapi juga berfungsi sebagai relay untuk node yang lain. Dengan kata lain, setiap node bekerja sama untuk membangun dan mengirimkan data di jaringan.

Konsekuensinya sangat dahsyat, antara lain:

- Kita dapat membangun jaringan yang *reliable*.
- Biaya instalasi sangat murah.
- Adaptif terhadap perubahan kondisi/topologi jaringan.
- Cocok untuk keperluan militer, wilayah perkampungan, pegunungan.

Sebuah jaringan mesh dapat dirancang menggunakan teknik *flooding* atau menggunakan teknik *routing*. Jika menggunakan teknik routing, *message* akan dikirim melalui sebuah jalur dengan cara "loncat" dari satu node ke node lain sampai tujuan tercapai. Untuk menjamin keberadaan *route/path,* maka sebuah mekanisme routing harus memungkinkan untuk terjadi sambungan terus-menerus dan mengkonfigurasi secara otomatis jika ada jalur yang rusak atau terblokir, menggunakan algoritma *self-healing* atau "memperbaiki diri sendiri". Sebuah jaringan mesh terjadi di mana semua node tersambung satu sama lain sebagai sebuah network yang saling tersambung.

Jaringan mesh dapat dilihat sebagai salah satu jenis jaringan *ad-hoc*. Mobile ad hoc network (MANET) dan jaringan mesh berhubungan sangat erat satu sama lain, hanya saja MANET harus dapat menyelesaikan masalah yang terjadi karena mobilitas node.

Kemampuan *self-healing* memungkinkan sebuah jaringan yang berbasis routing untuk tetap bekerja jika salah satu node rusak atau sambungan menjadi jelek. Akibatnya, jaringan ini umumnya sangat *reliable*. Biasanya ada lebih dari satu sambungan antara sumber dan tujuan di jaringan. Meskipun skenario ini biasanya digunakan di wireless, konsep ini juga dapat digunakan di jaringan kabel maupun interaksi software.

JARINGAN WIRELESS MESH

Wireless mesh networks awalnya dikembangkan untuk aplikasi militer yang memang mempunyai arsitektur mesh. Sepuluh tahun belakangan ini ukuran, biaya dan daya dari radio sudah sedemikian baik sehingga memungkinkan semakin banyak radio dapat bergabung sebagai node mesh.

Tambahan radio dalam setiap node memungkinkan dia untuk mendukung banyak fungsi, seperti akses bagi klien, layanan *backbone*, dan *scanning* (dibutuhkan untuk *handover* kecepatan tinggi di aplikasi mobile). Ditambah lagi, dengan semakin kecil radio, semakin murah biaya, semakin kecil daya yang dibutuhkan membuat node mesh semakin modular. Sebuah node atau sebuah device dapat berisi beberapa card radio atau modul, memungkinkan node untuk meng-*custom* fungsi dan frekuensi khusus.

CONTOH JARINGAN MESH DI DUNIA

Di awal tahun 2007, sebuah perusahaan Amerika, Meraki, meluncurkan sebuah mini wireless mesh router. Ini adalah contoh wireless mesh network yang diklaim dapat bekerja sampai 50Mbps. WiFi radio 802.11 di Meraki Mini dioptimasi untuk melakukan komunikasi jarak jauh, sampai jarak 250 meter. Ini adalah contoh sebuah jaringan mesh single-radio yang digunakan dalam sebuah komunitas sebagai saingan dari jaringan mesh multi-radio jarak jauh seperti BelAir atau MeshDynamics yang memberikan Infrastruktur multifungsi.

Naval Postgraduate School, Monterey CA, mendemonstrasikan jaringan wireless mesh untuk keamanan perbatasan. Dalam sistem percobaan ini, kamera di angkasa (yang diangkat menggunakan balon) mengirim video resolusi tinggi ke personal di bawah menggunakan jaringan mesh.

Proyek MIT Media Lab mengembangkan XO-1 laptop atau OLPC (*One Laptop per Child*) yang ditujukan untuk sekolah di negara berkembang menggunakan jaringan mesh (berbasis standar IEEE 802.11s) untuk membangun infrastruktur yang tahan banting tetapi murah. Sambungan yang dibangun oleh laptop diklaim oleh proyek OLPC dapat mengurangi kebutuhkan akan infrastruktur eksternal seperti internet untuk mencapai semua wilayah karena node yang saling tersambung dapat men-share sambungan ke node tetangga Konsep yang mirip diimplementasikan oleh Greenpacket dengan aplikasi yang disebut with its SONbuddy.

Di Cambridge, UK, pada tanggal 3 Juni 2006, sebuah jaringan mesh digunakan di mesh networking "Strawberry Fair" untuk menjalankan televisi live mobile, juga layanan radio dan Internet ke sekitar 80.000 orang.

Proyek Champaign-Urbana Community Wireless Network (CUWiN) mengembangkan software jaringan mesh berbasis pada implementasi open source dari Hazy-Sighted Link State Routing Protocol dan Expected Transmission Count Metric. Selanjutnya, Wireless Networking Group di University of Illinois at Urbana-Champaign mengembangkan multichannel, multi-radio wireless mesh testbed, dikenal dengan Net-X sebagai implementasi *proof of concept* dari *multichannel protocol* yang dikembangkan oleh grup tersebut. Implementasi ini berbasis pada sebuah arsitektur yang memungkinkan radio untuk berpindah kanal guna memelihara sambungan jaringan, termasuk di dalamnya protokol untuk alokasi channel dan routing.

SMesh adalah sebuah jaringan mesh wireless 802.11 multi-hop yang dikembangkan oleh Distributed System and Networks Lab di Johns Hopkins University. Sebuah skema *handoff* yang cepat memungkinkan *client mobile* untuk *roam* di jaringan tanpa terputus sambungan, sebuah fitur yang sangat dibutuhkan untuk aplikasi *real-time*, seperti VoIP.

Banyak jaringan mesh bekerja pada beberapa band sekaligus. Contoh, Firetide and Wave Relay Mesh Networks mempunyai pilihan untuk komunikasi antar node di 5.2 GHz atau 5.8 GHz, tetapi komunikasi dari node ke klien di 2.4 GHz (802.11). Hal ini dilakukan menggunakan SDR (*Software-Defined Radio*).

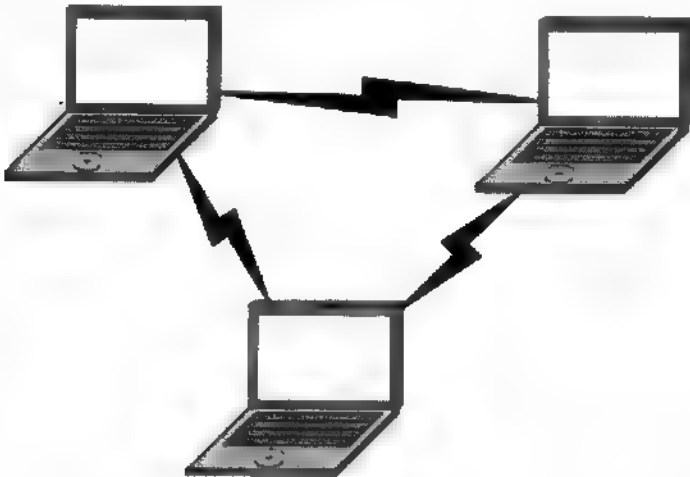
Proyek SolarMESH meneliti kemungkinan untuk menggunakan jaringan mesh berbasis 802.11 menggunakan tenaga matahari dan baterai *re-chargeable*. Access Point 802.11 biasa tampaknya kurang baik karena membutuhkan daya terus-menerus. Usaha standarisasi IEEE 802.11s mulai melihat kemungkinan pilihan untuk power save, tetapi penggunaan tenaga matahari pada node

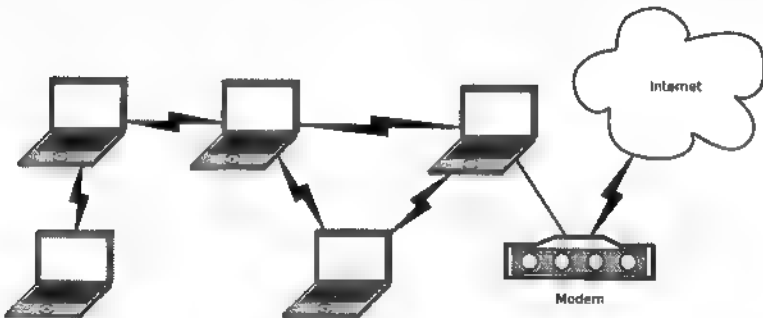
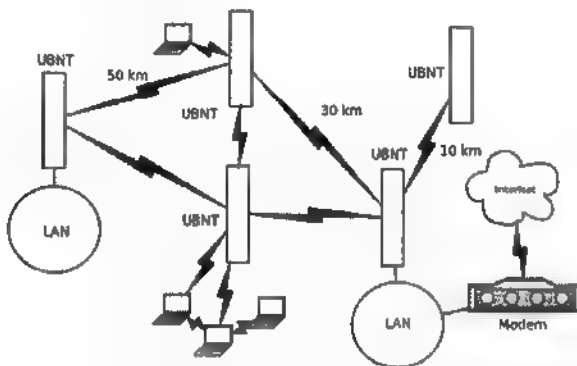
satu radio sebagai sambungan/relay tidak memungkinkan untuk melakukan penghematan daya (*power saving*).

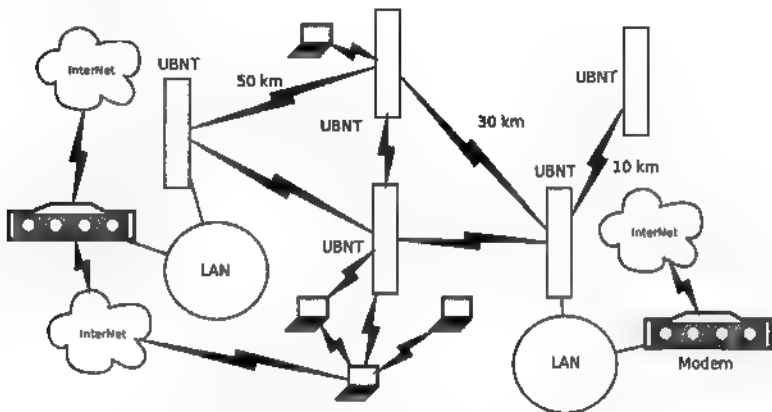
Proyek WING disponsori oleh Italian Ministry of University and Research dan dipimpin oleh CREATE-NET dan Technion mengembangkan algoritma dan protokol canggih yang memungkinkan jaringan wireless mesh sebagai standard arsitektur bagi Internet masa datang. Fokus utama diberikan pada masalah interferensi dan alokasi channel yang sensitif terhadap traffic jaringan, mendukung multi-radio/multi-interface dan penjadwalan oportunistik serta penggabungan traffic di lingkungan yang sangat tidak stabil.

Standar pada komunikasi kabel yang mulai dikembangkan mulai mengadopsi konsep jaringan mesh. Contoh ITU-T G.hn, sebuah standar untuk jaringan kecepatan tinggi (sampai 1 Gigabit/s) *local area network* menggunakan kabel listrik (*power lines*, kabel telepon dan kabel *coax*). Di lingkungan yang sangat bising seperti kabel PLN (di mana sinyal banyak diredam dan dikacaukan oleh *noise*) sangat biasa bila suatu device mengalami kesulitan untuk melihat device lain. Dalam situasi ini satu node dapat berfungsi sebagai relay dan meneruskan message untuk node yang tidak saling melihat, secara efektif ini membuat jaringan mesh. Di G.hn, relay dilakukan pada lapisan data link.

BEBERAPA SKENARIO JARINGAN MESH

Asal muasal jaringan mesh adalah jaringan wireless menggunaan mode ad-hoc di mana masing-masing komputer/laptop tersambung satu sama lain tanpa melalui Access Point, karena menggunakan wireless mode ad-hoc dan IP address statik. Ini sangat mudah dilakukan tanpa perlu tambahan software apa-apa.

Jaringan wireless ad-hoc menjadi rumit pada saat salah satu atau beberapa laptop tidak bisa mendengar satu sama lain. Konsekuensinya, kita perlu membuat mekanisme relay antar laptop agar secara keseluruhan satu sama lain dapat tersambung. Di sini mulai dikembangkan protokol-protokol yang memungkinkan setiap komputer/laptop mengetahui siapa tetangganya dan dapat memberitahukan seluruh komputer/laptop di seluruh jaringan. Ini adalah contoh sederhana sebuah jaringan wireless mesh. Salah satu protokol yang akan dijelaskan dalam buku ini adalah OLSR.

Jaringan akan menjadi lebih seru pada saat salah satu dari komputer/laptop tersambung ke Internet. Kita dapat mengkonfigurasi OLSR agar komputer yang tersambung dapat memberitahu semua komputer lain di jaringan MESH bahwa ada sambungan ke jaringan lain/Internet.

Sejauh ini skenario yang dikembangkan hanya mencakup jarak yang relatif pendek menggunakan komputer/laptop yang dilengkapi WiFi standard untuk jarak beberapa puluh meter. Bayangkan jika kita membangun jaringan menggunakan peralatan WiFi seperti Ubiquity/UBNT yang mampu menjangkau jarak 50-100 km. Kita dapat dengan mudah membangun jaringan untuk sebuah desa, pulau (seperti Ambon, Ternate, dll), juga wilayah rural yang luas seperti Papua.

Gambar di atas memperlihatkan sebuah topologi jaringan yang kompleks menggunakan Ubiquity (UBNT) yang sudah diubah firmware-nya agar mampu beroperasi menggunakan MESH. Jarak antar UBNT sangat jauh. Jika dikonfigurasi dengan baik bisa mencapai beberapa puluh km. Jadi kita dapat membangun jaringan backbone yang lumayan reliable menggunakan UBNT dalam konfigurasi MESH. Jika salah satu sambungan bermasalah, traffic akan secara otomatis pindah ke backbone lain yang masih baik sambungannya.

Kita juga dapat mengkonfigurasi UBNT atau device di jaringan MESH untuk memberitahukan akan keberadaan LAN yang tersambung di belakangnya. Dalam gambar ada beberapa LAN di jaringan.

Untuk tingkat yang lebih kompleks lagi, dalam jaringan MESH dimungkinkan ada beberapa gateway/sambungan ke Internet baik melalui router maupun salah satu laptop/komputer. Device mesh terdekat pada gateway tersebut dapat diprogram untuk memberitahu ke seluruh jaringan akan keberadaan sambungan ke Internet tersebut.

Yang tidak kalah menarik, salah satu router yang ada (router paling kiri) mempunyai beberapa sambungan ke Internet sekaligus. Program yang perlu diinstalasi di OpenWRT adalah multiwan agar mampu melakukan proses *load balancing* dengan beberapa sambungan ke Internet sekaligus.

BAB 2 TEKNIK ROUTING UNTUK MESH

Ada beberapa teknik protokol routing yang dapat digunakan untuk jaringan Mesh wireless. Protokol routing ini antara lain:

- Optimized Link State Routing Protocol (OLSR)
- Adaptive Wireless Path Protocol
- B.A.T.M.A.N.
- Babel (protocol)
- Open Shortest Path First (OSPF)
- Opportunity Driven Multiple Access (ODMA) – untuk selular network.
- ZigBee

Dari sekian banyak protokol yang ada, tampaknya yang banyak dipakai oleh komunitas Mesh di dunia adalah

- Optimized Link State Routing Protocol (OLSR)
- B.A.T.M.A.N.

Komunitas jaringan Mesh terbesar di dunia adalah FreiFunk yang beralamat di

http://freifunk.net

Pada kesempatan ini kita akan fokus pada Optimized Link State Routing Protocol (OLSR) yang merupakan protokol Mesh yang lumayan banyak diimplementasikan di dunia.

OPTIMIZED LINK STATE ROUTING PROTOCOL

OLSR daemon adalah sebuah implementasi dari protokol Optimized Link State Routing yang memungkinkan sebuah device jaringan untuk melakukan mesh routing.

OLSR bisa dijalankan, jika card WiFi mendukung mode *ad-hoc* dan juga pada device ethernet biasa. OLSR adalah salah satu dari dua standard Internet untuk jaringan Mesh. OSLR banyak digunakan dan sudah teruji keandalannya.

Karena OLSR bekerja di lapisan/layer 3, dia sangat portable. Saat ini dapat dijalankan di

- Windows (XP and Vista, Windows 7)
- Linux (i386, arm, alpha, mips, xscale)
- OS X (powerpc, intel, xscale, iPhone)
- VxWorks
- NetBSD
- FreeBSD
- OpenBSD
- Nokia N900
- Google phone (Android, G1)
- Linux Wifi Phones (WIP)

- Laptop
- Intel Classmate

OLSR sangat cepat dan menggunakan sedikit CPU time sehingga dapat menghemat baterai dari *embedded* dan *portable device*.

OLSR sangat *scalable.* OLSR dijalankan di jaringan komunitas wireless mesh dengan 2000 nodes (Athens wireless network), ~ 600 nodes (berlin FreiFunk.net), Leipzig Freifunk net, ~ 400 nodes (FunkFeuer.at). Sebagai protokol mesh routing yang dioperasikan di lapangan, OLSR sering menerima stress test dan cukup solid sejak versi 0.5.6-r7 terlepas dari apa yang dikatakan oleh protokol lain. Tentunya masih banyak ruang untuk perbaikan.

OLSR di-release menggunakan lisensi BSD, oleh karenanya OLSR sangat mudah untuk dimasukkan ke proyek Anda. Terima kasih untuk lisensi yang sangat liberal.

OLSR adalah sebuah proyek open source. Sangat disarankan Anda bergabung & ikut aktif mengembangkannya.

OLSR adalah routing protocol untuk jaringan mobile ad-hoc. Protokol tersebut bersifat pro-active, table driven dan menggunakan teknik *multipoint relaying* untuk *optimized message flooding.* Olsrd juga menjalankan link *quality extension*.

Olsrd dibuat dengan cara terstruktur dan terimplementasi dengan baik, sehingga mudah di-maintain, di-expand dan di-porting ke platform lain. Olsrd mempunyai arsitektur plugin yang sangat fleksibel.

Mengimplementasikan RFC3626 baik untuk core maupun fungsi tambahannya.

Olsrd mendukung penggunaan plugin yang dapat di-load. Hal ini dapat digunakan untuk menangani paket khusus yang dapat di-bawa oleh skema OLSR MPR flooding atau fungsi lain yang diinginkan.

FITUR YANG SPESIFIK DI OLSR

Protokol link-state routing seperti Open Shortest Path First (OSPF) dan IS-IS akan memilih router tertentu di setiap sambungan untuk melakukan pemberitahuan akan informasi topologi. Dalam jaring-an wireless ad-hoc ada perbedaan pengertian tentang sebuah sambungan. Paket dapat masuk dan keluar melalui interface yang sama. Oleh karena itu kita perlu melakukan proses yang berbeda untuk mengoptimasi proses pemberitahuan.

Menggunakan message Hello di setiap node akan menemukan informasi tetangga 2-hop dan melakukan pemilihan secara terdis-tribusi kumpulan dari multipoint relay (MPR). Node akan memilih MPR supaya ada jalur antara tetangga 2-hop melalui sebuah node yang terpilih sebagai MPR. MPR ini kemudian akan mengambil dan meneruskan message TC yang berisi MPR terpilih. Fungsi MPR membuat OLSR unik dari protokol link state routing dalam bebe-rapa hal: informasi path forwarding untuk message TC tidak dibagi ke semua node, tetapi bervariasi tergantung source. Hanya sebuah subset dari informasi source link state node, tidak semua link dari node diberitahukan tetapi hanya node yang merepresen-tasikan MPR yang terpilih saja.

Karena link-state routing membutuhkan database topologi yang tersinkronisasi di seluruh jaringan, OSPF dan IS-IS menyebarkan informasi topologi menggunakan algoritma yang reliable. Algorit-ma demikian sangat sulit diimplementasikan dalam jaringan wire-less ad-hoc. Oleh karenanya OLSR tidak peduli masalah reliabilitas,

cukup melakukan penyebaran informasi topologi sesering mungkin untuk memastikan bahwa database tidak sampai tidak terupdate dalam periode yang lama.

Keuntungan

Karena menggunakan protokol yang proaktif, route ke semua tujuan dalam jaringan akan diketahui dan dipelihara sebelum digunakan. Karena route akan tersedia dalam tabel routing standar menjadikan sangat berguna bagi beberapa sistem/aplikasi jaringan karena tidak ada delay untuk mencari route.

Overhead/waktu yang dibutuhkan untuk membuat routing, walaupun biasanya agak lebih lama daripada reactive protocol, tidak bertambah dengan jumlah route yang dibuat.

Route default dan network dapat dimasukkan ke sistem menggunakan message HNA yang memungkinkan sambungan ke InterNet atau jaringan lain dalam OLSR MANET cloud. Route network adalah salah satu kelebihan proactive protocol yang saat ini tidak bisa ditangani oleh reactive protocol dengan baik.

Nilai timeout dan informasi validitas ada dalam message yang memungkinkan nilai timer yang berbeda digunakan untuk berbagai node.

Kritik Terhadap OLSR

Definisi awal dari OLSR tidak termasuk kemampuan untuk mengetahui kualitas sambungan; OLSR hanya mengasumsikan sambungan terjadi, jika ada beberapa hello paket yang diterima. Hal ini mengasumsikan, bahwa sebuah sambungan hanya ada dua kemungkinan (jalan atau mati), yang sebetulnya belum tentu untuk jaringan wireless, yang kadang-kadang menunjukkan paket loss

yang tinggi. Oleh karena itu implementasi open source OLSRd (di Linux-based mesh router) mulai dikembangkan (mulai versi 0.4.8) untuk mendeteksi kualitas sambungan.

Karena proactive protocol, OLSR banyak menggunakan daya dan jaringan untuk menyebarkan data tentang route yang mungkin tidak digunakan. Hal ini mungkin bukan masalah bagi Access Point, atau laptop, akan tetapi ini menyebabkan OLSR tidak baik untuk jaringan sensor yang berusaha untuk tidur sepanjang waktu. Untuk Access Point kecil dengan CPU power yang rendah, proyek open source OLSRd menunjukkan bahwa jaringan mesh skala besar dapat dijalankan dengan OLSRd di ribuan node dengan CPU power yang sangat kecil di-embedded device 200MHz.

Karena protokol link-state, OLSR memerlukan bandwidth dan CPU power yang lumayan untuk menghitung jalur yang optimal di jaringan. Dalam jaringan yang mengandalkan OLSR, biasanya tidak lebih dari beberapa ratus node, tampaknya hal ini bukan masalah.

Dengan hanya menggunakan MPR untuk membanjiri informasi topologi, OLSR akan menghilangkan beberapa redudansi dalam proses penyebaran informasi. Hal ini mungkin akan menjadi masalah, jika dioperasikan dalam jaringan yang mempunyai paket loss yang lumayan. Akan tetapi mekanisme MPR mempunyai kemampuan untuk memperbaiki diri sendiri, yang artinya jika ada paket loss maka dimungkinkan ada beberapa node yang tidak ingin mengirim ulang paketnya..

MESSAGE DI OLSR

OLSR menggunakan message "Hello" untuk mencari tetangga satu hop dan tetangga dua hop melalui respons mereka. Pengirim kemudian dapat memilih multipoint relay (MPR) berbasis pada node satu hop yang memberikan route terbaik ke node dua hop. Setiap

node juga merupakan pemilih MPR dan akan memberitahukan node yang dia pilih sebagai node MPR. OLSR menggunakan message topology control (TC) bersama MPR forwarding untuk memberitahukan informasi tetangga ke seluruh jaringan. Message Host and Network Association (HNA) digunakan oleh OLSR untuk menyebarkan pemberitahuan network route sama seperti message TC memberitahukan route host.

Hello Paket

ANSN (bit 0–15)	Reserved (bit 16–31)
Advertised Neighbor Main Address	
Advertised Neighbor Main Address	

Topology Control (TC) Paket

PENDEKATAN LAIN

Masalah routing di jaringan wireless ad-hoc saat ini sedang aktif diteliti dan OLSR adalah salah satu dari solusi yang diusulkan. Bagi banyak orang, masih tidak jelas apakah kita perlu mengembangkan sebuah protokol yang baru atau OSPF dapat dikembangkan untuk interface wireless.

Dalam lingkungan rakus-bandwidth dan rakus-daya, kita biasanya tidak menginginkan jaringan yang ribut juga tidak ada traffic yang harus dilalukan. Protocol routing reactive biasanya tidak menyimpan tabel route, tetapi menyimpannya sesuai kebutuhan. Protokol link-state biasanya membutuhkan sinkronisasi database. Protokol ini biasanya menggunakan pendekatan distance vector, seperti AODV dan DSDV atau pendekatan yang lebih ad-hoc yang belum tentu dapat membangun jalur yang optimal, seperti Dynamic Source Routing.

OLSR VERSI 2

OLSRv2 saat ini mulai dikembangkan di IETF. OLSRv2 tetap menjaga banyak fitur kunci dari OLSRv1 termasuk pemilihan MPR dan diseminasi. Kunci perbedaan ada pada fleksibilitas dan desain modular menggunakan komponen yang digunakan bersama: paket format, paket bb dan protokol untuk mencari tahu tetangga NHDP. Komponen ini dirancang agar sama dengan protokol yang digunakan di IETF MANET. Perbedaan dalam menangani banyak alamat dan node dengan banyak interface juga tampak antara OLSR dan OLSRv2.

IMPLEMENTASI

- OLSR.ORG – tersedia source code OLSR di GNU/Linux, Windows, Mac OS X, FreeBSD dan NetBSD. Fitur dokumentasi yang lumayan lengkap, termasuk informasi survai pekerjaan yang terkait.
- NRL-OLSR – tersedia source code dari NRL-OLSR. Dapat dijalankan di Windows, MacOS, Linux, dan berbagai embedded PDA sistem seperti Arm/Zaurus dan PocketPC

termasuk simulator ns2 dan OPNET., **http://cs.itd.nrl.navy.mil/focus/**

- SOURCEFORGE.NET-OLSR – Dibuat oleh MOVIQUITY berbasis pada studi yang dilakukan oleh Project Workpad, menyediakan source code dalam C# untuk menjalankan MANET (Ad-Hoc, Meshnet) menggunakan protocol OLSR. Dikembangkan menggunakan WM 6, Win XP dapat diadaptasi ke platform lain menggunakan NET Framework dan Compact **http://sourceforge.net/projects/wmolsr/**

INSTALASI OLSR DI KOMPUTER

INSTALASI OLSR DI UBUNTU

Cara ini tidak terlalu disarankan. Jika kita ingin mencobanya, maka kita dapat melakukannya menggunakan perintah berikut:

```
sudo apt-get install olsrd olsrd-plugins
```

Hanya saja ini biasanya bukan versi bleeding edge :), tapi lumayan stabil untuk *ngoprek.*

Masalah dengan olsrd hasil apt-get adalah:

- olsrd dengan/etc/init.d/olsrd mencar/etc/olsrd.conf bukan /etc/olsrd/olsrd.conf. Akibatnya akan abort, karena tidak menemukan file konfigurasi.
- olsrd akan melaporkan error, buang LinkQualityDijkstraLimit dan LinkQualityWinSize di olsrd.conf.

Caranya edit

```
vi /etc/olsrd.conf
```

Remark line 124

```
# LinkQualityDijkstraLimit 3 3.0
```

Remark line 240

```
# LinkQualityWinSize    100
```

COMPILE DARI SOURCE CODE DI UBUNTU

Cara ini lebih disarankan dan biasanya menghasilkan aplikasi yang stabil.

Pertama-tama, siapkan aplikasi pendukung

```
sudo apt-get install kernel-package libncurses5-dev fakeroot wget \
bzip2 g++ libssl-dev libxml2-dev doxygen bison flex libc6
```

Download source code dari:

http://www.olsr.org/?q=download

Copy, extract dan compile:

```
cp olsrd-0.6.3.tar.bz2 /usr/local/src/
cd /usr/local/src
tar jxvf olsrd-0.6.3.tar.bz2
cd /usr/local/src/olsrd-0.6.3
make all
make install
make libs
make install_libs
mkdir -p /etc
cp -i files/olsrd.conf.default.lq /etc/olsrd.conf
```

Sedikit catatan tentang olsrd:

```
========= C O N F I G U R A T I O N - F I L E ============
olsrd uses the configfile /etc/olsrd.conf
a default configfile. A sample RFC-compliance aimed
```

configfile can be found in olsrd.conf.default.rfc.

However none of the larger OLSRD using networks use that

so install a configfile with activated link quality exstensions

per default.

can be found at files/olsrd.conf.default.lq

==

mkdir -p /etc

cp -i files/olsrd.conf.default.lq /etc/olsrd.conf

Edit /etc/olsrd.conf before running olsrd!!

Edit Konfigurasi OLSR

Edit olsrd.conf

```
sudo cp /etc/olsrd.conf /etc/olsrd.conf.original
sudo vi /etc/olsrd.conf
```

Pastikan ada interface yang aktif menggunakan OLSR, misalnya eth1, wlan0 dll. Contoh konfigurasi sederhana:

```
InterfaceDefaults {
   Ip4Broadcast     255.255.255.255
}
Interface "eth1" {
  # Ip4Broadcast 255.255.255.255
  # Ip6AddrType site-local
```

```
# Ip6MulticastSite ff05::11
# Ip6MulticastGlobal ff0e::1

HelloInterval 6.0
HelloValidityTime 600.0
TcInterval 0.5
TcValidityTime 300.0
MidInterval 10.0
MidValidityTime 300.0
HnaInterval 10.0
HnaValidityTime 300.0
# Weight 0
# LinkQualityMult 192.168.0.1 0.5
}
```

Tambahkan untuk Plugin txtinfo:

```
LoadPlugin "olsrd_txtinfo.so.0.1"
{
PlParam "port" "8080"
PlParam "Host" "127.0.0.1"
# PlParam "Net" "0.0.0.0 0.0.0.0"
# PlParam "Host" "80.23.53.22"
# # PlParam "Net" "192.168.0.0 255.255.0.0"
}
```

Kadang kala pada saat kita jalankan olsrd akan melaporkan error, buang LinkQualityDijkstraLimit dan LinkQualityWinSize di olsrd. conf. Caranya, edit:

```
vi /etc/olsrd.conf
```

Remark line 124:

```
# LinkQualityDijkstraLimit 3 3.0
```

Remark line 240:

```
# LinkQualityWinSize    100
```

Cek Konfigurasi OLSR

Walaupun ini biasanya tidak diperlukan. Untuk berjaga-jaga, sebaiknya cek konfigurasi menggunakan perintah

```
sudo olsrd -d 2
```

atau bagi yang menginstall menggunakan apt-get bisa

```
sudo olsrd -d 2 -f /etc/olsrd/olsrd.conf
```

Sepertinya menggunakan apt-get ada beberapa error. Beri # saja di line yang error. Biasanya setelah diberi # error yang ada maka olsrd akan jalan.

Edit Interface

Kita dapat melakukannya menggunakan

- iwconfig di console
- edit /etc/network/interface

Dengan asumsi eth1 adalah interface wireless. Edit di shell/ console

```
iwconfig eth1 mode ad-hoc
```

```
iwconfig eth1 essid MESH

iwconfig eth1 ap 02:ca:ff:ee:ba:be

iwconfig eth1 channel 1

ifconfig eth1 172.16.1.2

ifconfig eth1 netmask 255.255.0.0
```

atau

```
iwconfig eth1 mode ad-hoc essid MESH ap 02:ca:ff:ee:ba:be channel 1

ifconfig eth1 172.16.1.2 netmask 255.255.0.0
```

Edit:

```
vi /etc/network/interfaces
```

Isi kira-kira sebagai berikut:

```
auto lo

iface lo inet loopback


auto eth1

allow-hot plug eth1

iface eth1 inet static

address 172.16.1.2

broadcast 172.16.255.255

netmask 255.255.0.0

wireless-mode ad-hoc

wireless-channel 1
```

```
wireless-essid MESH
wireless-bssid 02:ca:ff:ee:ba:be

auto wlan0
allow-hot plug wlan0
iface wlan0 inet static
address 10.10.10.11
netmask 255.255.255.0
wireless-mode ad-hoc
wireless-channel 11
wireless-key s:12345
wireless-essid mesh

#auto eth0
#iface eth0 inet dhcp
```

Restart Networking

Networking dan OLSR:

```
sudo /etc/init.d/networking restart
sudo /etc/init.d/olsrd start
```

Jika pada saat melakukan

```
sudo /etc/init.d/network restart
```

keluar error berikut:

```
Error for wireless request "Set Mode" (8B06) :
SET failed on device wlan0 ; Invalid argument.
Error for wireless request "Set Mode" (8B06) :
SET failed on device wlan0 ; Device or resource busy.
```

Kemungkinan wireless interface/hardware yang Anda gunakan tidak mendukung mode ad-hoc. Kita perlu menggunakan USB wireless card lain.

Verifikasi Jaringan

Asumsinya eth1 adalah interface WLAN.

```
ifconfig eth1

eth1   Link encap:Ethernet  HWaddr 00:21:00:50:a3:90
inet addr:172.16.1.2  Bcast:172.16.255.255  Mask:255.255.0.0
inet6 addr: fe80::221:ff:fe50:a390/64 Scope:Link
UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
RX packets:52 errors:0 dropped:0 overruns:0 frame:188
TX packets:88 errors:11 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:3356 (3.3 KB)  TX bytes:14387 (14.3 KB)
Interrupt:17
```

Cek wireless

```
iwconfig eth1
```

```
eth1   IEEE 802.11bg  ESSID:"MESH"
Mode:Ad-Hoc  Frequency:2.412 GHz  Cell: 02:CA:FF:EE:BA:BE
Bit Rate=54 Mb/s   Tx-Power:24 dBm
Retry min limit:7   RTS thr:off   Fragment thr:off
Power Management:off
Link Quality=5/5  Signal level=-21 dBm  Noise level=-57 dBm
Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
Tx excessive retries:0  Invalid misc:0   Missed beacon:0
```

Cek Routing

Ketik

```
netstat -nr
```

Akan keluar kira-kira

```
Kernel IP routing table
Destination   Gateway      Genmask           Flags  MSS Window  irtt Iface
0.0.0.0       172.16.1.1   0.0.0.0           UG     0 0         0 eth1
172.16.0.0    0.0.0.0      255.255.0.0       U      0 0         0 eth1
172.16.1.1    172.16.1.1   255.255.255.255   UGH    0 0         0 eth1
172.16.1.2    172.16.1.2   255.255.255.255   UGH    0 0         0 eth1
```

Perhatikan host dengan netmask 255.255.255.255. Ini merupakan host yang ditambahkan secara otomatis oleh OLSR.

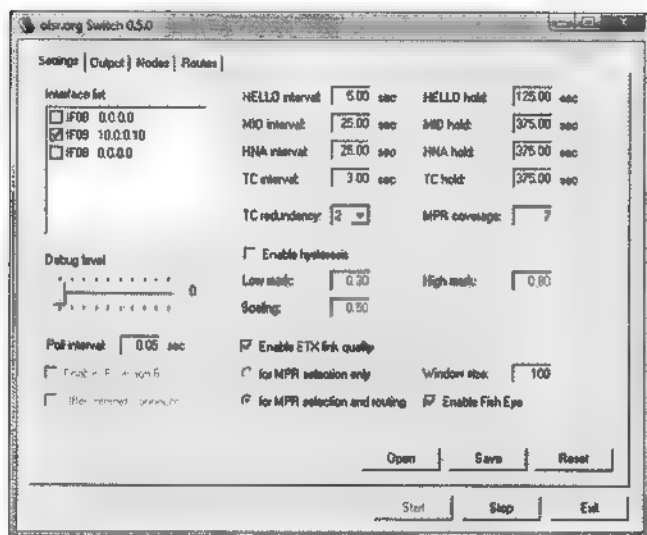
OLSR DI WINDOWS

Instalasi dan mengoperasikan OLSR di Windows sangat mudah.

- Set WiFi / interface wireless agar menjadi ad-hoc.
- Set IP statik.
- Download OLSR untuk Windows dari

  http://olsr.org/?q=download

  http://olsr.org/releases/0.5/olsrd-0.5.6-r3-pre-cac1df64dcd5-setup.exe

- Install seperti biasa. Klik **Exe, Next, Next, Next, Finish.**
- Jalankan.

- Klik pada Interface list, interface mana yang akan diaktifkan.
- Klik **Start.**

Pada tabel routing akan tampak, kira-kira:

FREIFUNK FRIMWARE

Bagi mereka yang ingin langsung mengoperasikan Mesh dengan router yang ada. Sebaiknya mencoba menggunakan Linksys WRT54GL dan FreiFunk Firmware. Sayang, saat ini Linksys WRT54GL sudah mulai langka.

KOMUNITAS MESH FREIFUNK

Freifunk (Bahasa Jerman untuk Free Radio) adalah sebuah inisiatif untuk mendukung pengembangan tools untuk jaringan free mesh. Di samping itu, inisiatif ini mendukung komunitas untuk mengembangkan pengetahuan untuk mensetup jaringan mereka sendiri.

Komunitas Freifunk adalah bagian dari proyek pengembangan tool jaringan mesh termasuk Freifunk Firmware dan OpenWrt projects, protokol routing seperti OLSR dan B.A.T.M.A.N., tool seperti untuk pemetaan jaringan (freimap), scanning tool seperti horst tool dan banyak lagi. Belakangan ini orang mulai mengembangkan Open Hardware seperti Mesh Potato untuk villagetelco.

Dengan freifunk firmware sebetulnya relatif mudah untuk membangun jaringan wireless mesh menggunakan WLAN ad-hoc pada lapisan komunikasi ke-2 dan ke-3 dengan OLSR, BATMAN atau protokol lain. Berasal dari Jerman, Freifunk telah berhasil mengimplementasikan Mesh di banyak negara. Proyek OLPC di Afganistan menggunakan Freifunk untuk implementasi mesh untuk mendistribusikan buku digital, berita maupun media pembelajar-

an. Di Ghana, Freifunk digunakan untuk menjembatani digital divide di perkampungan. Di Vietnam, Freifunk digunakan untuk memberikan sambungan internet di kegiatan Free and Open Source seperti FOSSASIA. Di Eropa dan di kota-kota dan perkampungan di Amerika, biaya sambungan jaringan untuk komunitas dan usaha kecil menengah diperingan dengan menshare akses Internet, seperti ADSL, telepon dan sambungan satelit ke jaringan remote.

FREIFUNK FIRMWARE/OPENWRT

Freifunk Firmware adalah versi linux OpenWRT yang dimodifikasi untuk router. Tujuan utama-nya agar firmware yang dibuat menjadikan mudah dan cepat dalam mensetup jaringan mesh wireless. Firmware dirancang untuk embedded linux untuk peralatan jaringan low power yang dapat diadaptasi ke berbagai device. Freifunk/OpenWrt mendukung komunikasi WLAN ad-hoc dan routing pada layer 2 dan 3. Protokol routing dapat diinstalasi melalui update paket. Protokol yang banyak digunakan untuk MESH adalah OLSR dan BATMAN. Dengan membuat Freifunk embedded firmware bersama tim OpenWRT, komunitas mengembangkan sebuah firmware yang dapat dikonfigurasi sendiri, memberbaiki diri sendiri dan men-tune secara automatis jaringan mesh (http://wiki.openwrt.org/Freifunk).

FITUR FREIFUNK FIRMWARE

Freifunk firmware memungkinkan semua orang dengan mudah membangun jaringan mesh dan router, *allows anyone to relatively easily build mesh networks of routers*. Jaringan ini secara mengejutkan dapat dikembangkan untuk melibatkan ratusan/ribuan node yang dibangun menggunakan router biasa (*off the shelf*).

Firmware PLUGIN

Versi test saat ini ada di SVN (https://dev.openwrt.org/wiki/GetSource) membuka kemungkinan developer untuk membuat plugin untuk firmware. Plugin ini memungkinkan user untuk memberikan layanan berbasis lokasi, pencarian lokal, router CMS, distribusi radio FM lokal melalui router, dan banyak lagi. Contoh dari layanan berbasis lokasi adalah Apple Bonjour yang memungkinkan pengguna untuk berbagi musik mereka di jaringan wifi lokal, akan tetapi melalui jaringan mesh FreiFunk. Masih banyak lagi cara untuk melakukan sharing dan berkomunikasi sebagai layanan yang dibuat di atas standard terbuka dan software open source.

LAYANAN LOKAL DI JARINGAN MESH

Proyek lain di mana kita mengintegrasikan layanan adalah proyek Villagetelco, di mana kita mengadaptasi firmware untuk layanan telepon melalui jaringan ad-hoc (www.villagetelco.org).

MENGADAPTASI FIRMWARE

Perkembangan ke arah yang lain, komunitas juga terlibat dalam mem-porting aplikasi maupun lingkungan desktop seperti LXDE ke dalam OpenWrt untuk Freifunk cloud untuk memberikan device yang cepat dengan kemampuan interface grafis X dan dapat memanfaatkan ribuan aplikasi yang tersedia. Porting dari code dari Google Android juga ditujukan untuk mendukung adaptasi dari Firmware untuk penggunaan tertentu.

KOMUNITAS DI DUNIA

Untuk membaca lebih lanjut tentang jaringan wireless yang bebas di seluruh dunia, silahkan kunjungi global FreiFunk newswire di http://global.freifunk.net.

Untuk memulai aktivitas, kita cukup melakukan dua hal, yaitu:

- Cek Firmware, download, dan install.
- Kontak komunitas lokal atau start komunitas yang baru.

LINKSYS WRT54GL UPGRADE FREIFUNK FIRMWARE

Download FreiFunk Firmware dari

- **http://ipkg.berlin.freifunk.net/_g+gl/**
- **http://ipkg.berlin.freifunk.net/_g+gl/openwrt-g-freifunk-1.7.4-en.bin**

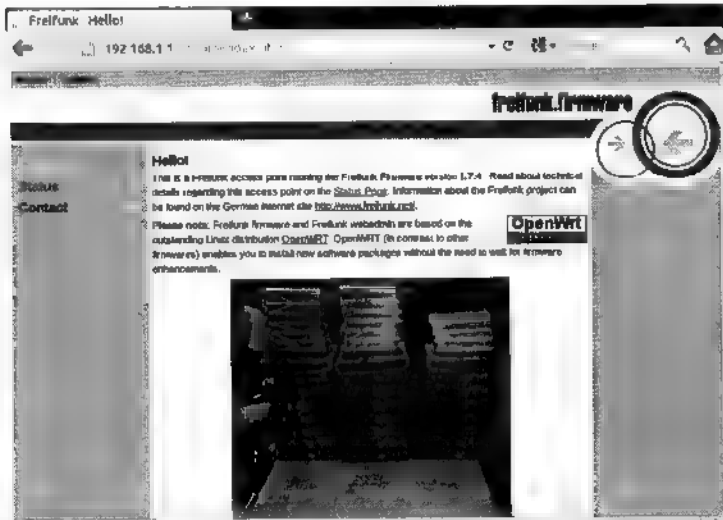
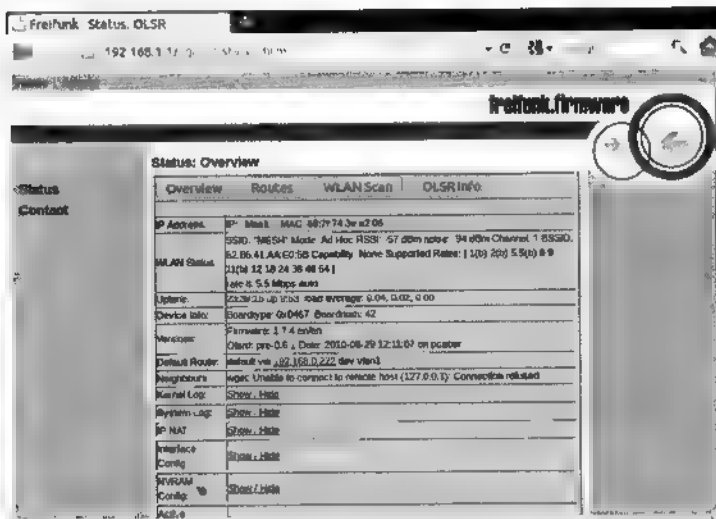
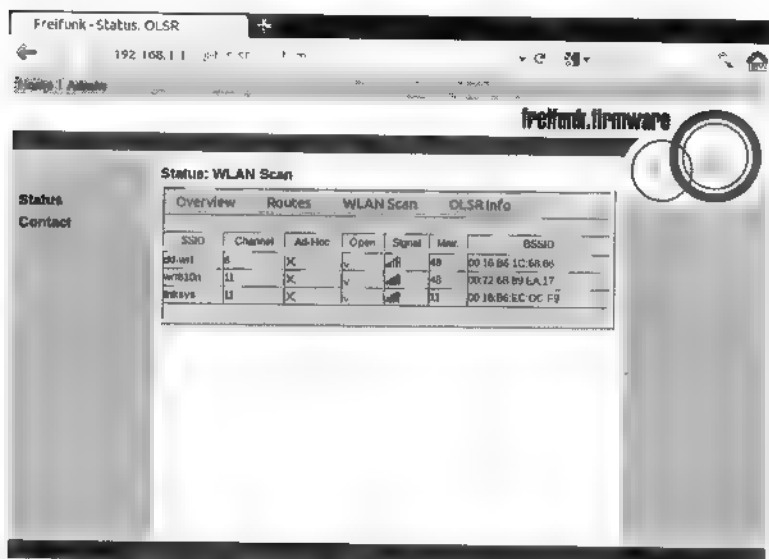
Upgrade melalui menu:

Administration -> Firmware Upgrade

Masukkan firmware openwrt-g-freifunk-1.7.4-en.bin

Jangan sampai sambungan putus, karena akan membuat Linksys Anda tewas dan tidak bisa digunakan sama sekali.

Setelah di-upgrade, kita dapat langsung masuk ke

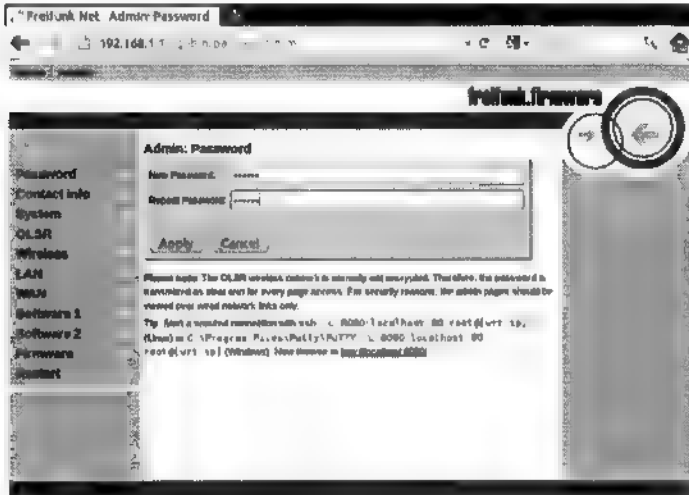
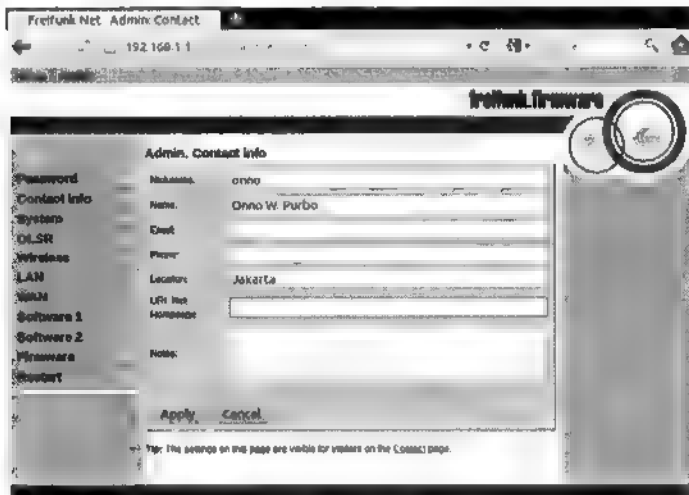
IP address	192.168.1.1
Username	root
Password	admin

Tampak pada gambar di atas adalah tampilan awal saat kita masuk ke Linksys WRT54GL yang sudah di-upgrade ke FreiFunk Firmware.

Jika kita masuk ke menu Status maka akan tampak status device yang kita gunakan. Seperti IP address, MAC Address, WLAN, Uptime, dll.

Jika kita klik **STATUS > WLAN SCAN** kita dapat melihat akses point yang sedang aktif di sekitar kita. Dalam contoh ada tiga (3) access point yang sedang aktif.

LINKSYS WRT54GL KONFIGURASI FREIFUNK

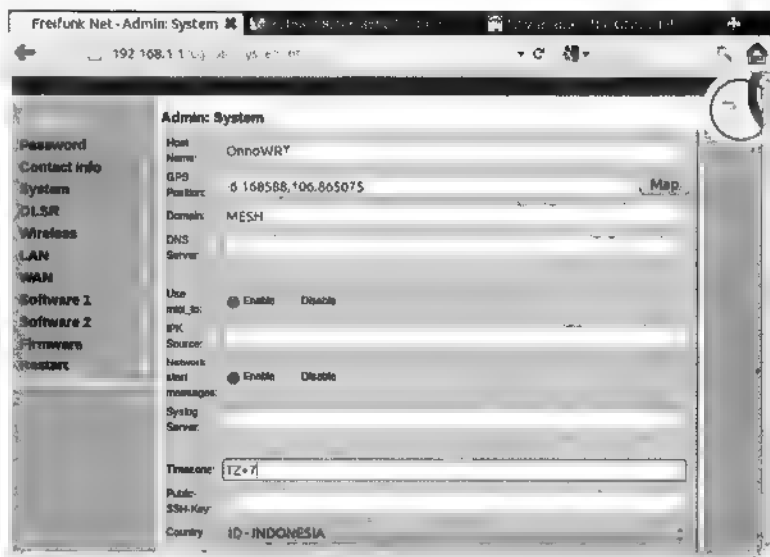
Pada bagian ini akan diperlihatkan tampilan konfigurasi awal FreiFunk Firmware.

Untuk konfigurasi awal FreiFunk, kita perlu masuk ke Admin. Di situ akan diperlihatkan "ToDo List" apa saja yang perlu kita lakukan.

Perhatikan kolom "State" dalam Todo-List. Yang disilang (X) adalah yang perlu dilakukan. Sementara yang di-V adalah yang sudah dilakukan.

Kita perlu menyelesaikan semua yang disilang (X).

Tampak pada tampilan di atas, saat kita men-set password admin. Biasanya kita harus mengulangi password yang kita ketik untuk memastikan, bahwa kita mengetik password yang sama.

Di bagian Admin: Contact Info, kita dapat memasukkan beberapa informasi seperti:

- Nickname
- Name
- E-mail
- Phone
- Lokasi
- URL
- Catatan lainnya

Sebaiknya isi semaksimal mungkin untuk memudahkan FreiFunk melihat seberapa besar jaringan komunitas mesh wireless di dunia.

Pada bagian Admin: System kita dapat memasukkan beberapa detail teknis yang penting, seperti:

Hostname

Posisi GPS, ini dapat di cari menggunakan Google Map & klik kanan pada lokasi kita.

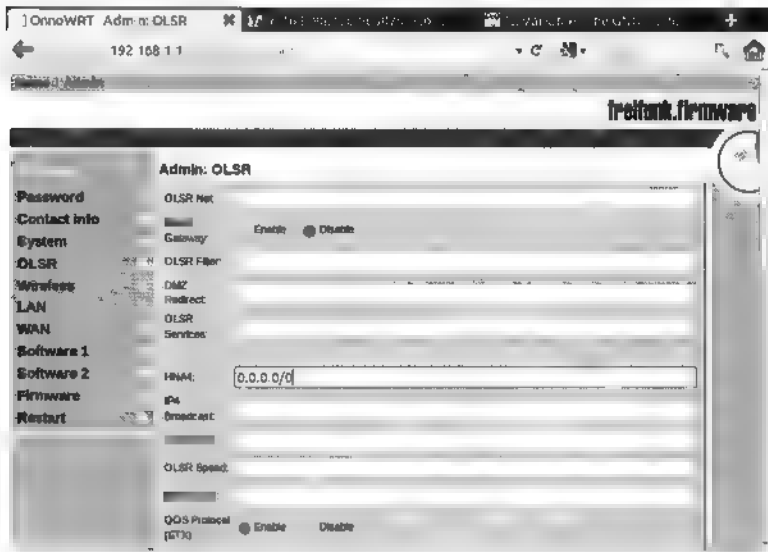
Domain

DNS Server

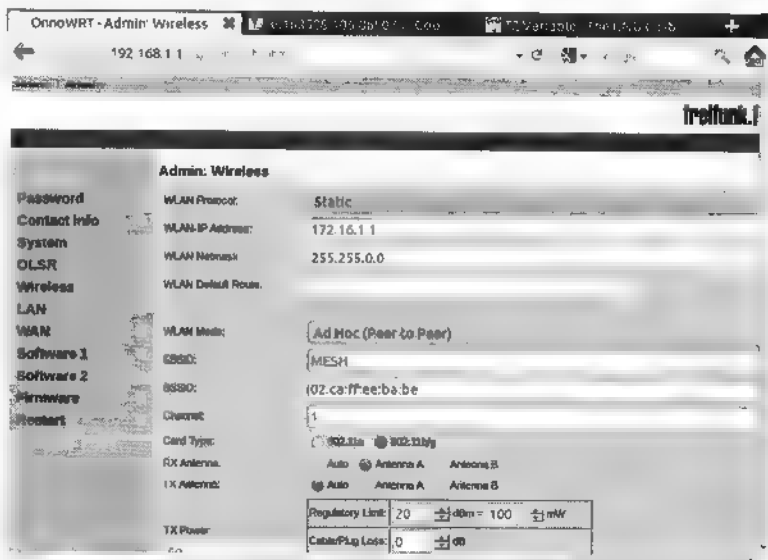
Time Zone (WIB adalah TZ+7).

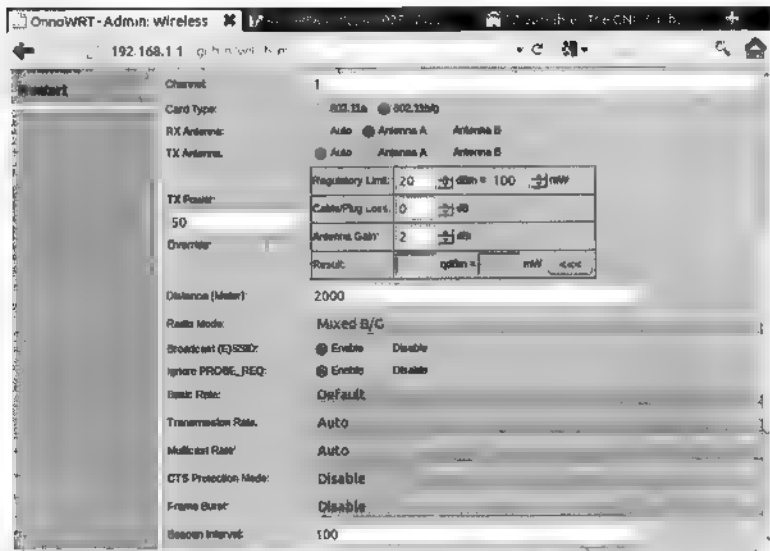
Negara (ID – Indonesia)

Linksys WRT54GL FreiFunk Firmware Konfigurasi OLSR

Konfigurasi OLSR sama sekali tidak sulit, bahkan sebetulnya hampir tidak perlu dikonfigurasi sama sekali untuk bisa operasional.

Pada menu Admin: OLSR sebetulnya sudah bisa operasional tanpa kita konfigurasi apa-apa. Jika kita ingin memberitahu jaringan bahwa:

- Internet dapat diakses melalui router yang kita gunakan. Kita cukup menambahkan pada HNA4 0.0.0.0/0.
- Jika kita ingin memberitahukan bahwa ada LAN lain di belakang kita, kita dapat menambahkan network address dari LAN tersebut, seperti 192.168.1.0/24 dsb.

Konfigurasi Linksys WRT54GL FreiFunk Firmware Wireless Setting

Melalui menu Admin: Wireless kita dapat menset interface wireless pada device Linksys WRT54GL. Beberapa parameter yang penting yang perlu di-set:.

WLAN Protocol	pastikan Static
WLAN IP address	sebaiknya keluar 10.x.x.x, tapi bisa juga 172.16.x.x.
WLAN Netmask	tergantung rencana jaringan Anda.
WLAN Mode	ad-hoc, pastikan jangan sampai menggunakan selain ad-hoc.
ESSID	terserah, saya biasanya menset menjadi MESH.
BSSID	biasanya teman-teman di FreiFunk menggunakan sesuatu yang bisa dibaca, misalnya 02:ca:ff:ee:ba:be
TX Power	set konservatif, misalnya 50mW.

Beberapa hal tambahan yang dapat di-set seperti:

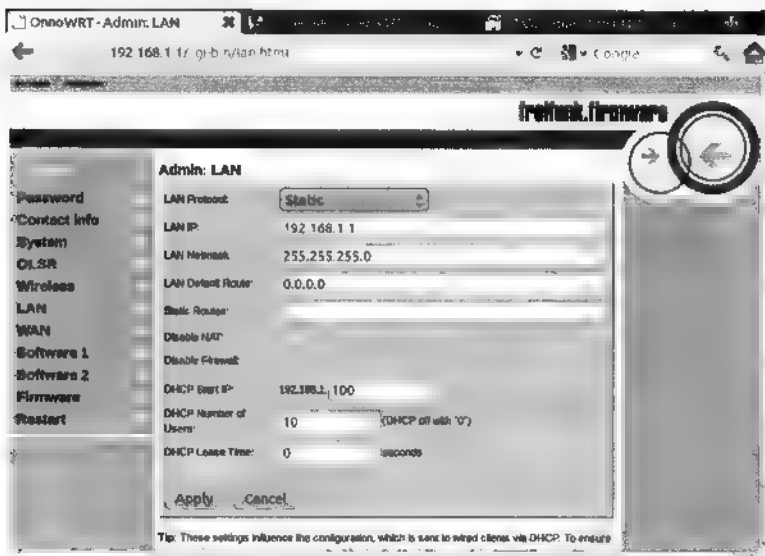
Jarak / distance yang dapat dijangkau oleh AP kita.

Radio Mode

Broadcast ESSID enable

dll.

Parameter-parameter terakhir ini dapat menggunakan setup default, tidak perlu diubah.

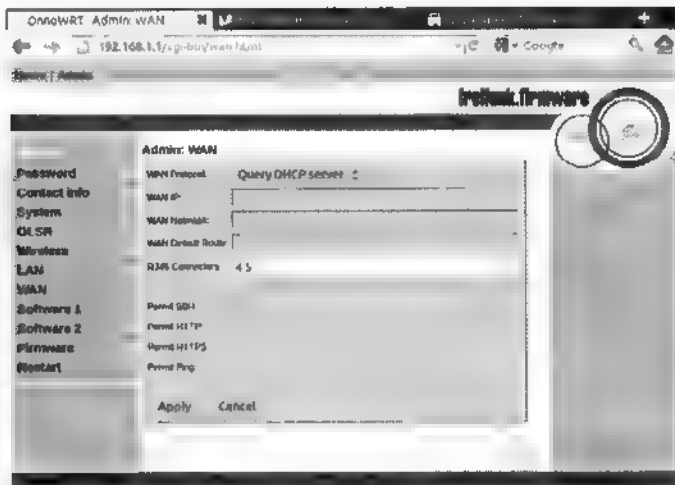
Konfigurasi Linksys WRT54GL FreiFunk Firmware LAN & WAN Setting

Pada Admin: LAN kita dapat men-set:

Konfigurasi IP LAN Statik

LAN IP set IP address LAN

LAN Netmask	biasanya 255.255.255.0
LAN Default route	biasanya 0.0.0.0
DHCP Start IP	terserah, bisa start dari 100
DHCP user	terserah, bisa 10

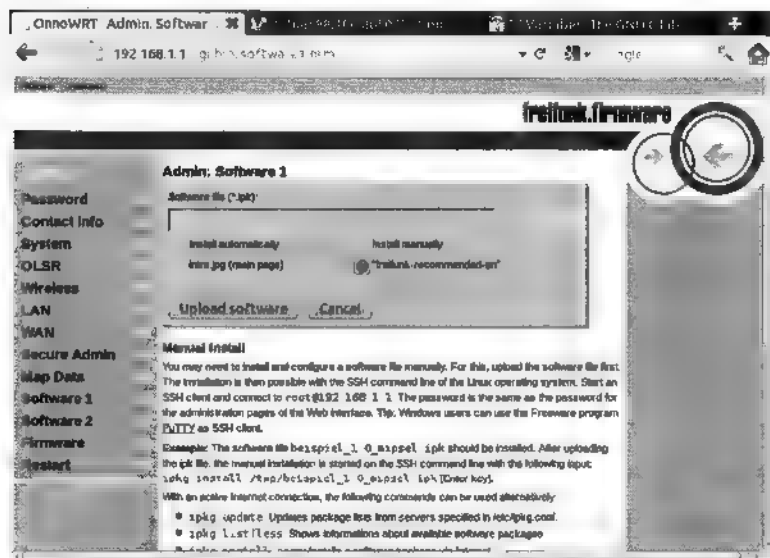
Pada konfigurasi LAN, jika Firewall di-enable, maka beberapa layanan seperti SSH dll akan terblokir. Jika kita ingin melalukan SSH dari laptop di-mesh, sebaiknya Firewall di-disable.

Jangan lupa klik Apply jika sudah di-set.

Selanjutnya konfigurasi Admin: WAN. Ini relatif sederhana. Yang lebih mudah adalah kita menggunakan

WAN Protocol	Query DHCP Server

Jika kita ingin mengijinkan user dari WAN untuk melakukan SSH, HTTP, HTTPS, ping, kita perlu meng-enable permit untuk layanan tersebut.

Klik **Apply** setelah selesai mengkonfigurasi.

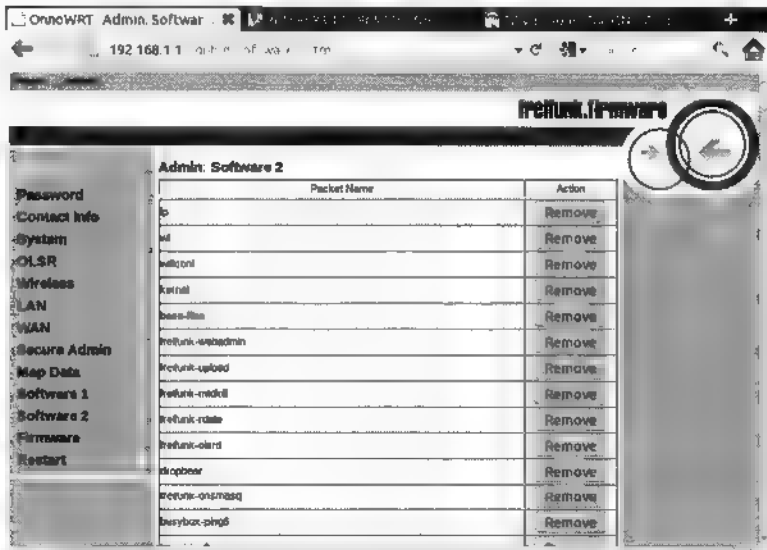
LINKSYS WRT54GL FREIFUNK FIRMWARE UPDATE & UPLOAD SOFTWARE

Melalui menu Admin: Software 1 kita dapat

- menginstall ipk
- jpg untuk main page
- instalasi software yang di rekomendasi FreiFunk

Yang perlu kita lakukan, isi kolom atau contreng apa yang ingin kita kerjakan, kemudian klik **Upload Software.**

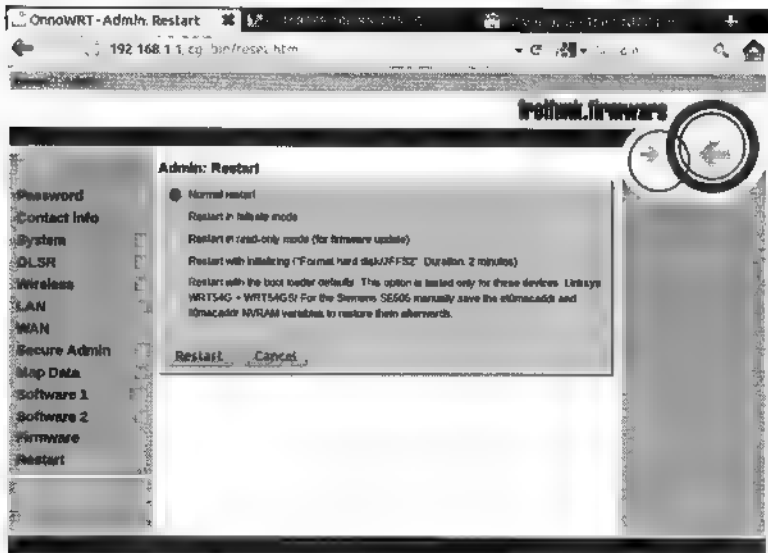
Pada gambar diperlihatkan tampilan saat kita menginstalasi software tambahan.

Perhatikan bahwa saat kita menginstalasi software tambahan, kita perlu memiliki sambungan internet yang baik. Software tambahan akan di-download sebelum di-install.

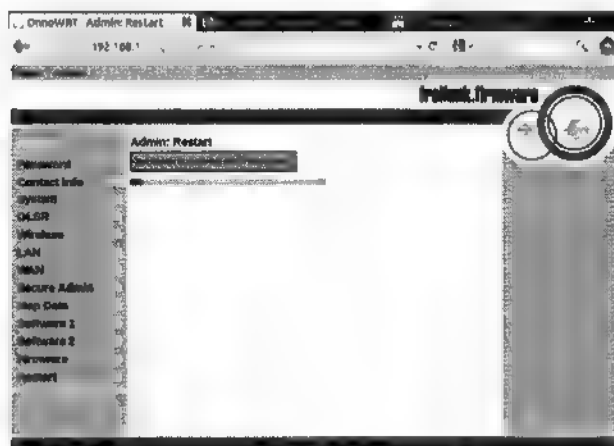
Kita perlu menunggu proses instalasi sampai semua selesai dan Done.

Melalui menu Admin: Software 2, kita dapat melihat daftar software aplikasi yang terinstalasi.

Jika software tersebut tidak dibutuhkan, kita dapat menghapusnya dengan meng-klik tombol **Remove** di sebelah kanan aplikasi yang dimaksud.

LINKSYS WRT54GL FREIFUNK FIRMWARE RESTART DEVICE

Kita dapat me-restart Linksys WRT54GL melalui menu Web.

Ada lima (5) pilihan restart, yaitu:

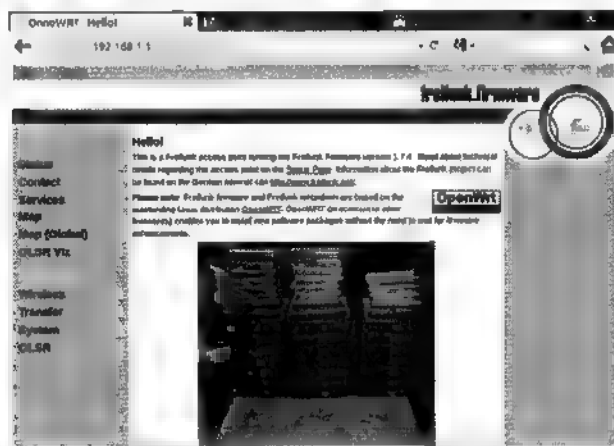
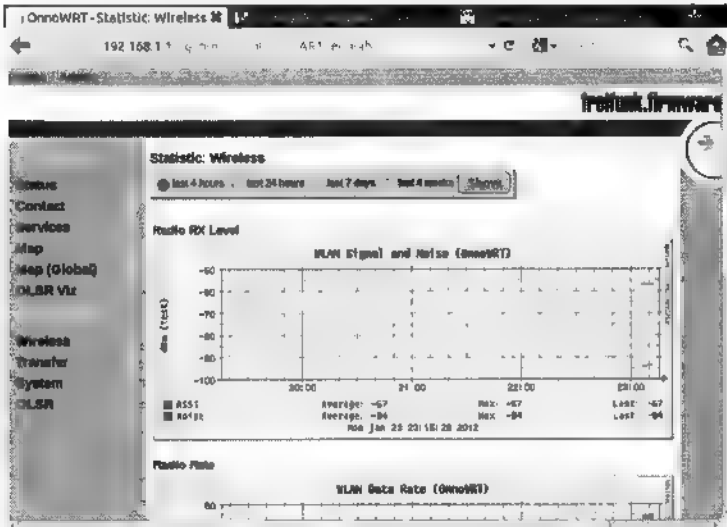
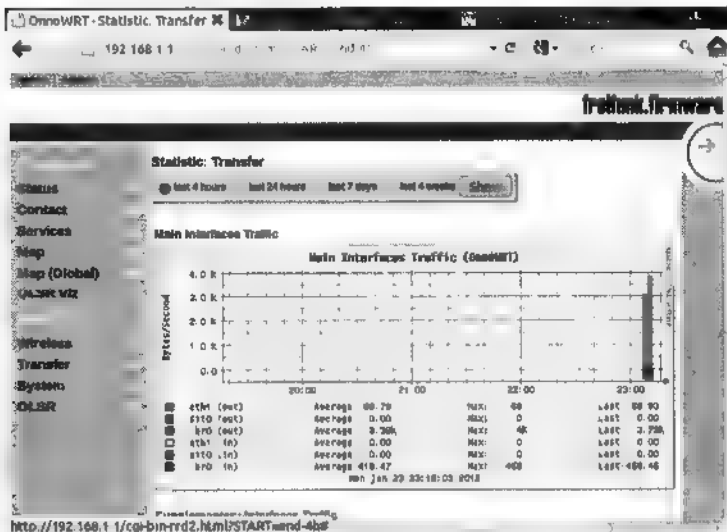
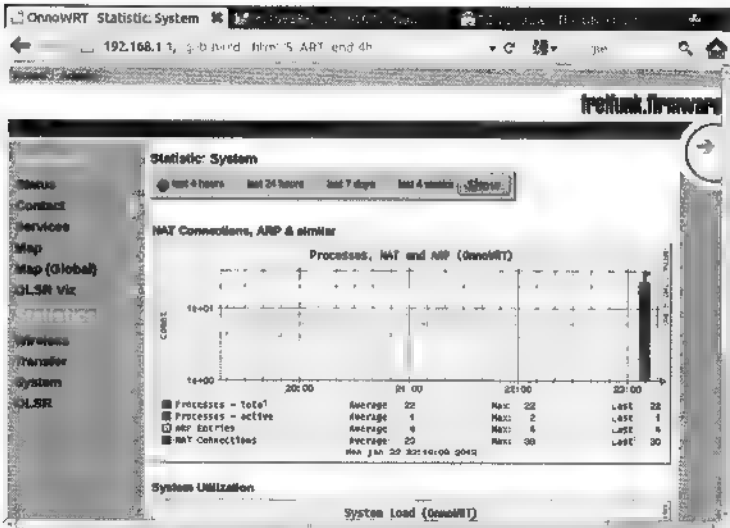
- Normal restart
- Restart dalam menu failsafe.
- Restart dalam mode read-only ini digunakan untuk firmware update.
- Restart untuk format harddisk.
- Restart dengan boot loader default.

Untuk operasi normal kita biasanya menggunakan normal restart.

Tampak pada gambar adalah tampilan saat Linksys WRT54GL restart.

TAMPILAN LINKSYS WRT54GL FREIFUNK FIRMWARE SETELAH UPGRADE APLIKASI

Setelah kita upgrade aplikasi, kita akan memperoleh beberapa tambahan aplikasi di linksys WRT54GL dengan FreiFunk Firmware.

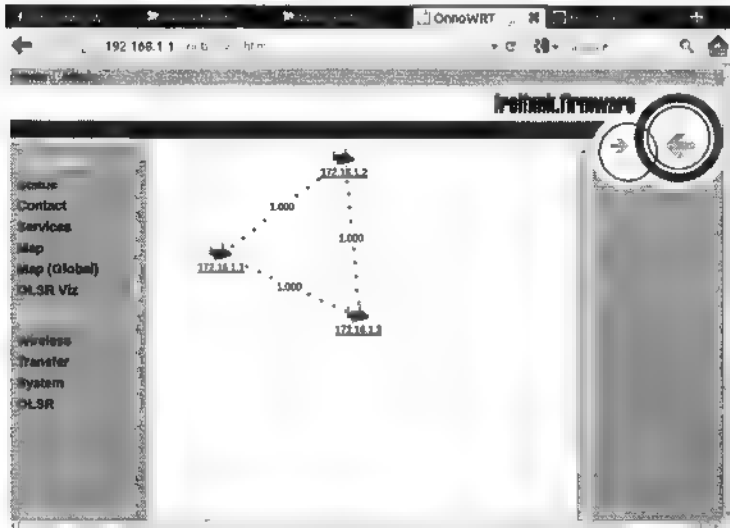
Tampak Home Page dengan tambahan aplikasi mempunyai beberapa tambahan menu yang tidak ada di Firmware awal.

Pada tampilan Status, di menu Neighbours terlihat ada parameter jika OLSR diaktifkan.

Pada tampilan map/peta, diperlihatkan posisi Access Point

Tampak pada gambar adalah tampilan Statistik Wireless. Kita dapat melihat grafik level sinyal, noise, kecepatan, dll.

Tampak pada gambar adalah tampilan Statistik Transfer Data. Kita dapat melihat grafik traffic masing-masing interface. Juga dapat dilihat traffic rata-rata, maksimum dan traffic terakhir.

Yang tidak kalah penting adalah tampilan statistik System. Kita dapat melihat dengan jelas,

- proses total
- proses yang aktif
- ARP
- sambungan NAT.
- Beban System
- dll.

Detail informasi seperti rata-rata (average), maksimum dan data terakhir juga dapat dilihat.

Bagi kita yang mengoperasikan MESH, yang penting juga adalah tampilan untuk melihat Status OLSR. Melalui Status OLSR kita dapat melihat dengan jelas:

- Sambungan (Links)
- Tetangga (Neighbours)
- Topologi jaringan
- Gateway ke jaringan lain, termasuk gateway ke Internet.
- Dll.

Yang akan memudahkan visualisasi jaringan MESH adalah OLSR Viz. Gambar di atas adalah visualisasi OLSR Viz dari tiga node mesh yang sedang beroperasi. Node 172.16.1.1, berwarna merah, juga memberikan layanan gateway ke Internet.

MEMBUAT SISTEM OPERASI OPENWRT

Bagi mereka yang kurang beruntung, tidak memperoleh Linksys WRT54GL, pilihan yang ada adalah mencari sistem operasi OpenWRT

OpenWrt dapat dilihat sebagai distribusi Linux untuk device embedded.

Daripada membuat sebuah firmware yang statik, OpenWRT menyediakan file system yang dapat ditulis dengan manajemen paket. Hal ini akan membebaskan kita dari vendor untuk bisa memilih dan mengkonfigurasi aplikasi, sehingga dapat mengkustomasi device melalui aplikasi yang cocok.

Untuk para Developer: OpenWRT adalah sebuah framework untuk membuat aplikasi tanpa pusing untuk membuat firmware-nya. Untuk user hal ini berarti kemungkinan untuk mengkustomasi secara penuh, untuk menggunakan device untuk hal-hal yang mungkin belum pernah dipikirkan sebelumnya.

DOWNLOAD FIRMWARE OPENWRT YANG SUDAH JADI

Bagi kita yang tidak mau mengambil risiko, kita dapat men-download firmware openwrt yang sudah jadi. Firmware yang sudah jadi biasanya mempunyai fasilitas standar, tetapi umumnya tidak

mempunyai fasilitas untuk mendukung MESH/FreiFunk. Mereka yang ingin mengoperasikan MESH mau tidak mau harus mengkompilasi firmware yang dibutuhkan.

Bagi mereka yang cukup dengan Firmware OpenWRT yang sudah jadi, langkah yang perlu dilakukan adalah:

- Masuk ke http://wiki.openwrt.org/toh/start cari device Anda, apakah masuk dalam daftar device yang didukung (support). Perlu dicatat bahwa tidak semua device didukung oleh OpenWRT.
- Masuk ke http://downloads.openwrt.org/snapshots/trunk/ untuk mengambil file .img dari firmware yang dimaksud. Tergantung chipset / processor yang digunakan. Contoh:
    - UBNT2 http://downloads.openwrt.org/snapshots/trunk/atheros/
    - UBNT http://downloads.openwrt.org/snapshots/trunk/ar71xx/
    - Buffalo http://downloads.openwrt.org/snapshots/trunk/ar71xx/

Sekali lagi, semua file .img yang disediakan oleh OpenWRT tidak mendukung MESH. Untuk operasi MESH ada dua pilihan, yaitu:

- Menambahkan aplikasi OLSR di firmware yang sudah jadi.
- Mengkompilasi sendiri sistem operasi OpenWRT.

Pada bagian ini akan dijelaskan cara mengkompilasi firmware sendiri. Sedangkan bagaimana menambahkan aplikasi OLSR di firmware yang sudah jadi akan diterangkan pada bagian selanjutnya.

PROSEDUR KOMPILASI FIRMWARE OPENWRT

Secara umum prosedur kompilasi firmware OpenWRT adalah:

- Lakukan pada komputer yang mempunyai harddisk free 30Gbyte.
- Yang aman menggunakan sistem operasi Ubuntu 10.10. Ubuntu 12.04 entah kenapa sering gagal compile.
- Lakukan sebagai non-root user.
- Lakukan semua perintah di <buildroot dir>, misalnya ~/openwrt/trunk/
- Update OpenWrt source
- Update & install package feeds.
- Configure build.
- Start build. Ini akan secara otomatis mengkompilasi Ttoolchain, cross-compile sources, package packages dan generate image siap untuk di-flashed.

Yang menarik di sini, kita mengkompilasi sebuah sistem operasi yang akan dijalankan di router yang menggunakan prosesor non-Intel. Jadi pada dasarnya kita harus melakukan proses cross compile dari komputer yang kita gunakan ke firmware yang nantinya akan dijalankan ke mesin non-Intel.

Persiapan Install aplikasi yang dibutuhkan untuk mengkompilasi firmware biasanya:

```
sudo su

apt-get update

apt-get install subversion gawk flex sharutils bison libz-dev patch fakeroot lzma \
```

```
git-core build-essential
```

Kita dapat men-download & mengkompilasi firmware menggunakan SVN sebagai user biasa:

```
svn co svn://svn.openwrt.org/openwrt/branches/backfire ~/openwrt/trunk/

cd ~/openwrt/trunk/

./scripts/feeds update

./scripts/feeds install <pkg_name_1> <pkg_name_2> ... <pkg_name_N>

make defconfig

make menuconfig

make world
```

Jika nanti perlu meng-update aplikasi dapat menggunakan perintah:

```
cd ~/openwrt/trunk/

svn update

./scripts/feeds update
```

Alternatif lain, kita dapat men-download menggunakan GIT

```
git clone git://nbd.name/openwrt.git ~/openwrt/trunk/

cd ~/openwrt/trunk/

./scripts/feeds update

./scripts/feeds install <pkg_name_1> <pkg_name_2> ... <pkg_name_N>

make menuconfig
```

```
make world
```

Jika nanti perlu mengupdate aplikasi dapat menggunakan perintah:

```
cd ~/openwrt/trunk/

git pull

./scripts/feeds update
```

BEBERAPA TIP SAAT KOMPILASI FIRMWARE OPENWRT

Menentukan Chipset

Kita dapat menentukan/menset chipset mana yang akan kita gunakan. Di target/linux/chipname/, ada makefile yang mengatakan kernel mana yang akan dibuat. Contohnya di Kamikaze, tepatnya kamikaze/target/linux/atheros/Makefile.

```
LINUX_VERSION:=2.6.23.17
```

Ubah ini sesuai dengan subversion yang Anda inginkan.

Build di Multi-core CPU

Proses build dapat dipercepat dengan menjalankan beberapa concurrent job process mengunakan option -j seperti:

```
make -j 3
```

Gunakan standard formula <jumlah CPU + 1>. Jika cara ini menyebabkan error saat build, coba compile tanpa option -j.

Build di Background

Jika Anda ingin menggunakan sistem/mesin Anda waktu Anda build, Anda dapat membuat proses build hanya menggunakan I/O dan kapasitas CPU yang idle, contoh untuk CPU dualcore:

```
ionice -c 3 nice -n 20 make -j 2
```

Build Single Packages

Saat kita melakukan developing atau packaging software untuk OpenWrt, akan lebih memudahkan, jika kita dapat membuat untuk package yang kita inginkan saja (misalnya untuk package cups):

```
make package/cups/compile V=99
```

Untuk package mc (*midnight commander*), yang kebetulan berada feed packages, perintahnya akan seperti:

```
make package/feeds/packages/mc/compile v=99
```

Melihat Build Errors

Jika build gagal, cara yang paling gampang untuk melihat error adalah melalui perintah

```
make V=99 2>&1 | tee build.log | grep -i error
```

Perintah di atas akan menyimpan copy full verbose dari output build (dengan stdout di pipe ke stderr) di /openwrt/trunk/build.log dan hanya memperlihatkan error saja ke screen

Contoh lain:

```
ionice -c 3 nice -n 20 make -j 2 V=99
CONFIG_DEBUG_SECTION_MISMATCH=y 2>&1 | tee build.log
| egrep -i '(warn|error)'
```

Cara di atas menyimpan copy full verbose dari output build (dengan stdout di pipe ke stderr) di build.log dan akan mengeluarkan hanya warning dan error saat building menggunakan resource background di dual core CPU.

Notifikasi Beep

Tergantung pada CPU yang kita gunakan, proses build akan memakan waktu sebentar atau lama. Jika kita menginginkan notifikasi suara, kita dapat menggunakan echo -e '\a':

```
make V=99 ; echo -e '\a'
```

Lokasi Image Firmware Hasil Compile

Jika berhasil, image yang kita buat akan berada di bawah direktori

<buildroot dir>/bin

Jika kita buat firmware untuk ar71xx maka ada di bawah

<buildroot dir>/bin/ar71xx directory.

Jika <buildroot dir> adalah /openwrt/trunk, maka binary ada di

/openwrt/trunk/bin/ar71xx.

Membersihkan/Clean Up

Kadang kita ingin membersihkan lingkungan build. Contoh berikut akan sangat berguna,

Delete isi bin dan directory build_dir

```
make clean
```

Dirclean

```
make dirclean
```

Delete isi /bin dan /build_dir directory juga /staging_dir dan /toolchain (=cross-compile tool). 'Dirclean' adalah untuk memperoleh "Full clean".

Distclean

```
make distclean
```

Hancurkan semua yang pernah kita compile atau configure dan delete semua content feed dan package source yang pernah kita download.

PERHATIAN: Perintah ini akan menghapus build configuration (.config).

Melihat Daftar Package

Melihat daftar package aplikasi yang dapat dimasukkan dalam firmware OpenWRT agak tricky. Informasi nama package dibutuhkan waktu proses compile Firmware dan dimasukkan dalam daftar di

```
./scripts/feeds install
```

Agak sulit mencarinya di web OpenWRT. Secara umum sebaiknya kita melihat di ipk yang sudah jadi. Misalnya di

http://downloads.openwrt.org/snapshots/trunk/ar71xx/

Beberapa paket yang penting, khususnya kalau mau membuat OpenWRT untuk OLSR:

meshwizard

openssh-server

openssh-client

luci

luci-app-meshwizard

luci-app-olsr

luci-app-olsr-services

luci-app-olsr-viz

luci-app-freifunk-policyrouting

luci-mod-freifunk

luci-theme-freifunk-generic

olsrd

olsrd-mod-txtinfo

olsrd-mod-dyn-gw

olsrd-mod-dyn-gw-plain

olsrd-mod-dot-draw

freifunk-common

freifunk-gwcheck

freifunk-mapupdate

CONTOH PROSES COMPILE OPENWRT FIRMWARE UNTUK BUFFALO WZR HP G450H

Perhatikan bahwa cara ini akan membutuhkan banyak akses ke Internet untuk download, termasuk waktu proses make. Termasuk download kernel linux, waktu compile toolchain :(... Tampaknya tidak bisa OFFLINE :(

Dalam contoh ini akan diperlihatkan cara compile firmware Buffalo WZR HP G450H untuk keperluan MESH dengan OLSRD. Oleh karenanya dibutuhkan aplikasi seperti

- luci
- olsrd
- freifunk

Pastikan

- Compile untuk AR71xx
- Profile WZR HP G450H
- Driver wireless Atheros AR9381? di install.
- Network olsrd

Siapkan pendukung aplikasi pendukung, lakukan

```
sudo su

apt-get update

apt-get install subversion gawk flex sharutils bison libz-dev patch fakeroot lzma \

git-core build-essential
```

Compile menggunakan development trunk sebagai user biasa

```
svn co svn://svn.openwrt.org/openwrt/trunk/ ~/openwrt.trunk/trunk

cd ~/openwrt.trunk/trunk/

./scripts/feeds update

./scripts/feeds install meshwizard openssh-server openssh-client

./scripts/feeds install freifunk-common freifunk-gwcheck freifunk-mapupdate
```

```
./scripts/feeds install luci luci-app-meshwizard luci-app-olsr luci-app-olsr-services

./scripts/feeds install luci-app-olsr-viz luci-app-freifunk-policyrouting luci-mod-freifunk

./scripts/feeds install luci-theme-freifunk-generic olsrd olsrd-mod-txtinfo olsrd-mod-dyn-gw

./scripts/feeds install olsrd-mod-dyn-gw-plain olsrd-mod-dot-draw

./scripts/feeds install olsrd-mod-arprefresh olsrd-mod-watchdog olsrd-mod-tas

./scripts/feeds install olsrd-mod-pgraph olsrd-mod-secure olsrd-mod-bmf olsrd-mod-httpinfo

./scripts/feeds install luci-app-upnp minidlna miniupnpc miniupnpd luci-app-minidlna rtorrent

./scripts/feeds install usb-modeswitch usb-modeswitch-data usbutils multiwan

./scripts/feeds install 6in4 6rd 4th 6to4 6tunnel libdaemon

make defconfig

make menuconfig

make -j 2 world
```

Setup Menuconfig

Menuconfig untuk Firmware Buffalo WZRHPG450H untuk OLSR dilengkapi dengan WebGUI LuCI, menggunakan modem 3G, load balancing untuk multiwan.

Target System > Atheros AR7xxx/AR9xxx

Subtarget > Generic

Target Profile > Buffalo WZR-HP-G450H

Target Images > tar.gz

Base system > bridge

Network > Routing and Redirection > ip

Network > Routing and Redirection > relayd

Network > Routing and Redirection > olsrd

Network > Routing and Redirection > olsrd >

olsrd-mod-arprefresh

olsrd-mod-bmf

olsrd-mod-dot-draw

olsrd-mod-dyn-gw

olsrd-mod-dyn-gw-plain

olsrd-mod-httpinfo

olsrd-mod-jsoninfo

olsrd-mod-mdns

olsrd-mod-nameservice

olsrd-mod-p2pd

olsrd-mod-pgraph

olsrd-mod-quagga

olsrd-mod-secure

olsrd-mod-tas

olsrd-mod-txtinfo

olsrd-mod-watchdog

Network > SSH > openssh-server

Network > Time Synchronization > ntpclient

Network > Web Servers/Proxies > uhttpd

uhttpd-mod-lua

uhttpd-mod-tls

uhttpd-mod-ubus

Network > hostapd

hostapd-mini

hostapd-utils

multiwan

samba36-client

samba36-server

tc

vnstat

vnstati

wshaper

Network >

chat

ppp

ppp-mod-pppoa

ppp-mod-pppoe

ppp-mod-pppol2tp

ppp-mod-pptp

ppp-mod-radius

ppp-multilink

pppdump

pppstats

LuCI > 1. Collections >

luci

luci-ssl

2. Modules >

luci-mod-freifunk

3. Applications >

luci-app-freifunk-policyrouting

luci-app-meshwizard

luci-app-multiwan

luci-app-ntpc

luci-app-olsr

luci-app-olsr-services

luci-app-olsr-viz

luci-app-p2pblock

luci-app-samba

luci-app-statistics

luci-app-vnstat

luci-app-wshaper

4. Themes >

luci-theme-bootstrap

luci-theme-freifunk-bno

luci-theme-freifunk-generic

6. Protocols >

luci-proto-3g

7. Server Interfaces >

luci-sgi-uhttpd

9. Freifunk >

luci-mod-freifunk-community

meshwizard

Kernel modules > Wireless Drivers >

kmod-ath9k-htc

kmod-ath9k-common

kmod-ath9k

Kernel modules > Network Support >

kmod-bridge

kmod-pptp

kmod-wprobe

Kernel Modules > USB Support >

kmod-usb-serial

kmod-usb-serial-ipw

kmod-usb-serial-option

kmod-usb-serial-qualcomm

kmod-usb-serial-wwan

kmod-usb-acm

kmod-usb-storage

kmod-usb-storage-extras

Utilities >

comgt

iwinfo

Mail > msmtp

msmtp-nossl

msmtp-queue

Sebetulnya flash/harddisk yang ada dalam Buffalo sangat besar. Kita dapat memasukkan banyak aplikasi ke dalamnya.

Jika diperlukan untuk update SVN, di folder trunk, lakukan:

```
cd ~/openwrt.trunk/trunk/
svn cleanup
svn update
./scripts/feeds update
```

Lokasi image firmware OpenWRT untuk Buffalo ada di

~/openwrt.trunk/trunk/bin/ar71xx/

Tepatnya

~/openwrt.trunk/trunk/bin/ar71xx/openwrt-ar71xx-generic-wzr-hp-g450h-squashfs-tftp.bin

CONTOH PROSES COMPILE OPENWRT FIRMWARE UNTUK UBNT

Cara ini akan membutuhkan banyak akses ke Internet untuk download, termasuk waktu proses make. Tampaknya tidak bisa OFFLINE :(

Dalam contoh ini akan diperlihatkan cara compile firmware Ubiquiti NanoStation2 untuk keperluan MESH dengan OLSRD. Oleh karenanya dibutuhkan aplikasi seperti

- olsrd
- meshwizard

Pastikan

- Compile untuk atheros
- Profile UBNT
- Driver wireless Atheros AR9381? di install.
- Network olsrd

Siapkan aplikasi pendukung untuk dapat mengkompilasi Firmware OpenWRT. Lakukan:

sudo su

apt-get update

apt-get install subversion gawk flex sharutils bison libz-dev patch fakeroot lzma \

git-core build-essential

Pada saat make menuconfig, kita secara umum kita dapat membuat dua (2) pilihan, yaitu,

- Membuat Firmware UBNT2 dengan OLSR dan LuCI WebGUI
- Membuat Firmwate UBNT2 dengan OLSR TANPA LuCI WebGUI ini khususnya untuk node di remote yang berfungsi hanya untuk relay saja.

Pilihan menuconfig untuk UBNT2 dengan OLSR dan LuCI WebGUI adalah sebagai berikut:

Target System > Atheros AR231x/AR5312

Target Profile > Default

Target Images > tar.gz

Base system > bridge

Network > Routing and Redirection > ip

Network > Routing and Redirection > relayd

Network > Routing and Redirection > olsrd

Network > Routing and Redirection > olsrd >

olsrd-mod-arprefresh

olsrd-mod-bmf

olsrd-mod-dot-draw

olsrd-mod-dyn-gw

olsrd-mod-dyn-gw-plain

olsrd-mod-httpinfo

olsrd-mod-jsoninfo

olsrd-mod-mdns

olsrd-mod-nameservice

olsrd-mod-p2pd

olsrd-mod-pgraph

olsrd-mod-quagga

olsrd-mod-secure

olsrd-mod-tas

olsrd-mod-txtinfo

olsrd-mod-watchdog

Network > SSH > openssh-server

Network > hostapd

Network > Time Synchronization > ntpclient

Network > Web Servers/Proxies > uhttpd

uhttpd-mod-lua

uhttpd-mod-tls

uhttpd-mod-ubus

LuCI > 1. Collections >

luci

luci-ssl

2. Modules >

luci-mod-freifunk

3. Applications >

luci-app-freifunk-policyrouting

luci-app-meshwizard

luci-app-ntpc

luci-app-olsr

luci-app-olsr-services

luci-app-olsr-viz

luci-app-p2pblock

luci-app-statistics

4. Themes >

luci-theme-bootstrap

luci-theme-freifunk-bno

luci-theme-freifunk-generic

7. Server Interfaces >

luci-sgi-uhttpd

9. Freifunk >

luci mod freifunk community

meshwizard

Kernel modules > Wireless Drivers >

kmod-ath9k-htc

kmod-ath9k-common

kmod-ath9k

Kernel modules > Network Support >

kmod-bridge

Utilities > iwinfo

Sementara, menuconfig untuk UBNT2 untuk OLSR TANPA WebGUI LuCI agar lebih ringan ke system adalah sebagai berikut,

Target System > Atheros AR231x/AR5312

Target Profile > Default

Target Images > tar.gz

Base system > bridge

Network > Routing and Redirection > ip

Network > Routing and Redirection > relayd

Network > Routing and Redirection > olsrd

Network > Routing and Redirection > olsrd >

olsrd-mod-arprefresh

olsrd-mod-bmf

olsrd-mod-dot-draw

olsrd-mod-dyn-gw

olsrd-mod-dyn-gw-plain

olsrd-mod-httpinfo

olsrd-mod-jsoninfo

olsrd-mod-mdns

olsrd-mod-nameservice

olsrd-mod-p2pd

olsrd-mod-pgraph

olsrd-mod-quagga

olsrd-mod-secure

olsrd-mod-tas

olsrd-mod-txtinfo

olsrd-mod-watchdog

Network > SSH > openssh-server

Network > hostapd

Network > Time Synchronization > ntpclient

Kernel modules > Wireless Drivers >

kmod-ath9k-htc

kmod-ath9k-common

kmod-ath9k

Kernel modules > Network Support >

kmod-bridge

Utilities > iwinfo

Untuk UBNT yang lain, khususnya untuk UBNT M, kita harus menggunakan Target System & Profile yang lain,

Target System > Atheros AR7xxx/AR9xxx

Subtarget > Generic

Target Profile > Ubiquiti Products

Base system > bridge

Network > Routing and Redirection > ip

Network > Routing and Redirection > relayd

Network > Routing and Redirection > olsrd

Network > Routing and Redirection > olsrd >

olsrd-mod-arprefresh

olsrd-mod-bmf

olsrd-mod-dot-draw

olsrd-mod-dyn-gw

olsrd-mod-dyn-gw-plain

olsrd-mod-httpinfo

olsrd-mod-jsoninfo

olsrd-mod-mdns

olsrd-mod-nameservice

olsrd-mod-p2pd

olsrd-mod-pgraph

olsrd-mod-quagga

olsrd-mod-secure

olsrd-mod-tas

olsrd-mod-txtinfo

olsrd-mod-watchdog

Network > SSH > openssh-server

Network > hostapd

Network > Time Synchronization > ntpclient

Network > Web Servers/Proxies > uhttpd

uhttpd-mod-lua

uhttpd-mod tls

uhttpd-mod-ubus

LuCI > 1. Collections >

luci

luci-ssl

2. Modules >

luci-mod-freifunk

3. Applications >

luci-app-freifunk-policyrouting

luci-app-meshwizard

luci-app-ntpc

luci-app-olsr

luci-app-olsr-services

luci-app-olsr-viz

luci-app-p2pblock

luci-app-statistics

4. Themes >

luci-theme-bootstrap

luci-theme-freifunk-bno

luci-theme-freifunk-generic

7. Server Interfaces >

luci-sgi-uhttpd

9. Freifunk >

luci-mod-freifunk-community

meshwizard

Kernel modules > Wireless Drivers >

kmod-ath9k-htc

kmod-ath9k-common

kmod-ath9k

Kernel modules > Network Support >

kmod-bridge

Utilities > iwinfo

Compile Firmware OpenWRT menggunakan Development Trunk, lakukan sebagai user biasa

```
svn co svn://svn.openwrt.org/openwrt/trunk/ ~/openwrt.trunk/trunk

cd ~/openwrt.trunk/trunk/

./scripts/feeds update

./scripts/feeds install meshwizard openssh-server openssh-client

./scripts/feeds install freifunk-common freifunk-gwcheck freifunk-mapupdate
```

```
./scripts/feeds install luci luci-app-meshwizard luci-app-olsr luci-app-olsr-services

./scripts/feeds install luci-app-olsr-viz luci-app-freifunk-policyrouting luci-mod-freifunk

./scripts/feeds install luci-theme-freifunk-generic olsrd olsrd-mod-txtinfo olsrd-mod-dyn-gw

./scripts/feeds install olsrd-mod-dyn-gw-plain olsrd-mod-dot-draw olsrd-mod-httpinfo

./scripts/feeds install olsrd-mod-arprefresh olsrd-mod-watchdog olsrd-mod-tas

./scripts/feeds install olsrd-mod-pgraph olsrd-mod-secure olsrd-mod-bmf

./scripts/feeds install 6in4 6rd 4th 6to4 6tunnel libdaemon

make defconfig

make menuconfig

make -j 2 world
```

Untuk UBNT2, lokasi Image Firmware OpenWRT hasil compile ada di,

```
~/openwrt.trunk/trunk/bin/atheros/
```

tepatnya file

```
~/openwrt.trunk/trunk/bin/atheros/openwrt-atheros-ubnt2-squashfs.bin
```

Untuk UBNTM, lokasi Image Firmware OpenWRT hasil compile ada di,

```
~/openwrt.trunk/trunk/bin/ar71xx/
```

tepatnya file

~/openwrt.trunk/trunk/bin/ar71xx/openwrt-ar71xx-generic-ubnt-nano-m-squashfs-factory.bin

INSTALASI DAN OPERASI OPENWRT PADA ROUTER DAN AP

Pada bagian ini akan dijelaskan proses instalasi dan operasi OpenWRT pada Router atau Access Point. Sebelum kita mem-flash image, mengubah sistem operasi bawaan sebuah router atau Access Point, sebaiknya kita mengetahui terlebih dahulu bagaimana mengembalikan sistem operasi ke sistem operasi awal. Istilah yang digunakan adalah Recovery Mode.

TEKNIK RECOVERY UBNT

Download firmware asli UBNT dalam extension .bin dari **http://www.ubnt.com/download.** Download sesuai dengan Device UBNT yang kita miliki.

Untuk UBNT, langkah yang perlu dilakukan adalah,

- Set IP address PC / laptop ke 192.168.1.x
- Sambungkan PC / laptop ke UBNT melalui kabel LAN yang baik. Pastikan sambungan baik, karena kalau sampai sambungan LAN putus saat tftp akan sangat fatal dan dapat membuat UBNT kita tewas.
- Matikan Device UBNT

- Tekan tombol Reset
- Nyalakan Device UBNT
- Lepaskan tombol Reset setelah ~10 detik (tidak lebih) sesudah device UBNT dinyalakan. Kita akan mengetahuinya, karena LED akan berubah.
- Ping 192.168.1.20. Jika ada respons, kita siap untuk meng-upload image. Jika tidak maka kita perlu mengulangi lagi dari awal.

Di sisi PC/laptop, lakukan:

- Matikan automatic DHCP pada laptop / PC
- Set IP statik pada 192.168.1.x
- cd ke folder tempat image

tftp image dari PC / laptop ke UBNT melalui kabel LAN menggunakan mode binary ke 192.168.1.20 sebagai 'flash_update'.

```
tftp 192.168.1.20
tftp> binary
tftp> trace
tftp> put NanoStation2-v4.0.1.build4978.bin flash_update
tftp> quit
```

Contoh di atas adalah untuk me-recovery NanoStation2 ke firmware asli yang dapat diambil di UBNT.com. Seperti,

http://www.ubnt.com/download#NanoStation2

http://www.ubnt.com/download#NanoStation2:Loco

http://www.ubnt.com/download#PicoStation2

Jika ingin mem-flash menggunakan OpenWRT kita dapat menggunakan perintah kira-kira sebagai berikut:

```
tftp 192.168.1.20
tftp> binary
tftp> trace
tftp> put openwrt-atheros-ubnt2-squashfs.bin flash_update
tftp> quit
```

Contoh di atas adalah untuk me-recovery NanoStation2 menjadi firmware OpenWRT

```
tftp 192.168.1.20
tftp> binary
tftp> trace
tftp> put openwrt-ar71xx-generic-ubnt-nano-m-squashfs-factory.bin flash_update
tftp> quit
```

Contoh di atas adalah untuk merecovery NanoStation M2 menjadi firmware OpenWRT.

Setelah berhasil di tftp upload, kita perlu men-cek apakah UBNT yang kita gunakan dapat berfungsi. Caranya, diamkan beberapa saat sambil di-ping ke IP address default

192.168.1.20 (UBNT)

192.168.1.1 (OpenWRT)

Setelah UBNT di-flash menggunakan OpenWRT, kita perlu tunggu sekitar 2-3 menitan supaya firmware betul-betul di-write ke flash. Kemudian lakukan,

telnet 192.168.1.1

Akan keluar kira-kira:

```
Trying 192.168.1.1...
Connected to 192.168.1.1.
Escape character is '^]'.
=== IMPORTANT ============================
 Use 'passwd' to set your login password
 this will disable telnet and enable SSH
------------------------------------------


BusyBox v1.19.4 (2012-09-04 06:16:13 WIT) built-in shell (ash)
Enter 'help' for a list of built-in commands.

  _______                     ________        __
 |       |.-----.-----.-----.|  |  |  |.----.|  |_
 |   -   ||  _  |  -__|     ||  |  |  ||   _||   _|
 |_______||   __|_____|__|__||________||__|  |____|
          |__| W I R E L E S S   F R E E D O M
-----------------------------------------------------
 ATTITUDE ADJUSTMENT (Bleeding Edge, r33315)
-----------------------------------------------------
  * 1/4 oz Vodka      Pour all ingredients into mixing
```

```
* 1/4 oz Gin       tin with ice, strain into glass.
* 1/4 oz Amaretto
* 1/4 oz Triple sec
* 1/4 oz Peach schnapps
* 1/4 oz Sour mix
* 1 splash Cranberry juice
-----------------------------------------------------
root@OpenWrt:/#
```

Ubah password root menggunakan perintah

```
passwd
(password root)
(password root)
exit
```

TEKNIK RECOVERY BUFFALO

Untuk memudahkan proses recovery Router Buffalo sebaiknya kita membuat script kecil yang dapat dijalankan secara cepat, misalnya:

```
vi buatflashbuffalo.sh
```

Untuk Buffalo WZR HP G450H isi, misalnya, dengan

```
ifconfig eth0:11 192.168.11.2
arp -s 192.168.11.1 02:aa:bb:cc:dd:23
curl -T openwrt-ar71xx-generic-wzr-hp-g450h-squashfs-tftp.bin tftp://192.168.11.1
```

Save script di tempat firmware openwrt tersebut berada. Jangan lupa

chmod 777 buatflashbuffalo.sh

Langkah yang perlu dilakukan untuk melakukan recovery adalah

- Matikan Buffalo
- Masukkan kabel LAN PC ke LAN4 Buffalo yang pas persis sebelah sambungan Internet.
- Masuk ke folder tempat image openwrt berada.
- Jadi superuser

  sudo su
- Masukan power Buffalo
- Secepatnya, dalam waktu kurang dari 4 detik, jalankan

  ./buatflashbuffalo.sh

Akan tampak pada layar

```
 % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                Dload  Upload   Total   Spent    Left  Speed
100 6272k    0     0  100 6272k    0   452k  0:00:13  0:00.13 --:--:--  796k
100 6272k    0     0  100 6272k    0   452k  0:00:13  0:00:13 --:--:--  452k
```

Jika kita cek arp -n akan tampak

```
arp -n
Address          HWtype   Hwaddress           Flags Mask        Iface
192.168 11 1     ether    02:aa:bb:cc dd 23   CM                eth0
```

Perintah ARP untuk beberapa device Buffalo

WZR HP AG300H	arp -s 192.168.11.1 02:aa:bb:cc:dd:20
WZR HP G300N	arp -s 192.168.11.1 aa:bb:cc:dd:ee·ff
WZR HP G300NH	arp -s 192.168.11.1 02:aa:bb:cc:dd.20

WZR HP G450H arp -s 192.168.11.1 02:aa:bb:cc:dd:23

Setelah Buffalo di-flash menggunakan OpenWRT, pada laptop lakukan

```
dhclient eth0
```

ping ke 192.168.1.1. Seharusnya mendapat respons balik dari ping yang kita lakukan.

Selanjutnya, telnet ke Buffalo pada IP address 192.168.1.1

```
telnet 192.168.1.1
```

Akan keluar,

```
Trying 192.168.1.1...
Connected to 192.168.1.1.
Escape character is '^]'.
 === IMPORTANT ============================
 Use 'passwd' to set your login password
this will disable telnet and enable SSH
------------------------------------------



BusyBox v1.19.4 (2012-09-05 17:24:46 WIT) built-in shell (ash)
Enter 'help' for a list of built-in commands.


  _______                     ________        __
 |       |.-----.-----.-----.|  |  |  |.----.|  |_
```

```
| - || _ | -_|  || | | || _|| _|
|______||  _|____|_|_||______||_| |___|
   |__| W I R E L E S S   F R E E D O M
-----------------------------------------------------
ATTITUDE ADJUSTMENT (Bleeding Edge, r33315)
-----------------------------------------------------
* 1/4 oz Vodka     Pour all ingredients into mixing
* 1/4 oz Gin       tin with ice, strain into glass.
* 1/4 oz Amaretto
* 1/4 oz Triple sec
* 1/4 oz Peach schnapps
* 1/4 oz Sour mix
* 1 splash Cranberry juice
-----------------------------------------------------
root@OpenWrt:/#
```

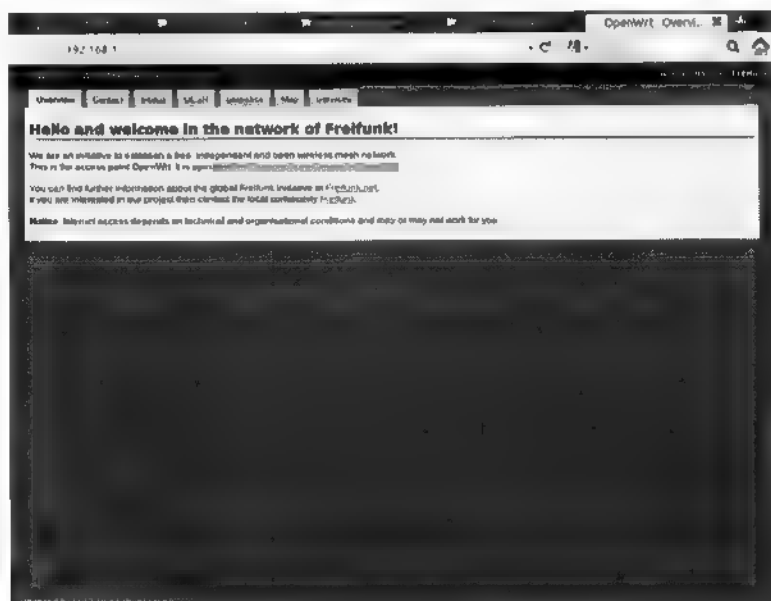
Ubah password root OpenWRT

```
passwd
(masukkan password root)
(masukkan password root)
exit
```

Selesai sudah proses aktivasi password root. Selanjutnya kita bisa mengakses Buffalo melalui Web atau melalui SSH.

SETUP IDENTITAS SAAT PERTAMA KALI MENGGUNAKAN OPENWRT

Pada saat pertama kali menggunakan, kita biasanya perlu menset identitas dari router OpenWRT. Hal ini akan menjadi informasi penting/bagian dari jaringan MESH FreiFunk nantinya. Tampilan saat kita mengisi data/informasi terlihat pada beberapa cuplikan gambar berikut:

OpenWrt - Basic
Hide OpenStreetMap
Contact
Reset
Save
Save & Apply

INSTALASI & OPERASI OLSR DI UBNT MENGGUNAKAN BACKFIRE OPENWRT

Bagi Anda yang tidak mengkompilasi sendiri firmware OpenWRT untuk UBNT, untuk menggunakan Backfire OpenWRT untuk UBNT2, perlu mengingat hal berikut.

Kebutuhan:

- Komputer dengan browser (mis. Firefox) dan ssh client.
- Ubiquity Device dengan OpenWRT yang sudah terinstalasi. Firmware dapat diambil dari

  http://downloads.openwrt.org/backfire/10.03.1/atheros/
- Install Firmware menggunakan Device Recovery UBNT.
- Akses Internet yang berlaku untuk Device & Computer. Edit network pada OpenWRT UBNT agar sesuai dengan jaringan,

Edit Network:

```
vi /etc/config/network
```

Pastikan:

```
config interface   lan
option ifname      eth0
option type        bridge
option proto       static
option ipaddr      192.168.0.10
option netmask     255.255.255.0
option gateway     192.168.0.222
```

Edit nameserver:

```
vi /etc/resolv.conf
```

Isi misalnya dengan:

```
nameserver 8.8.8.8
```

Untuk akses SSH ke UBNT, pastikan kita sudah menset root password. Akses ssh ke device ssh ke device, misalnya:

```
ssh root@192.168.1.1
```

Setelah login, lakukan update opkg.list

```
opkg update
```

Install luci administration UI & olsr show:

```
opkg install luci luci-app-olsr luci-app-olsr-viz luci-app-olsr-services
```

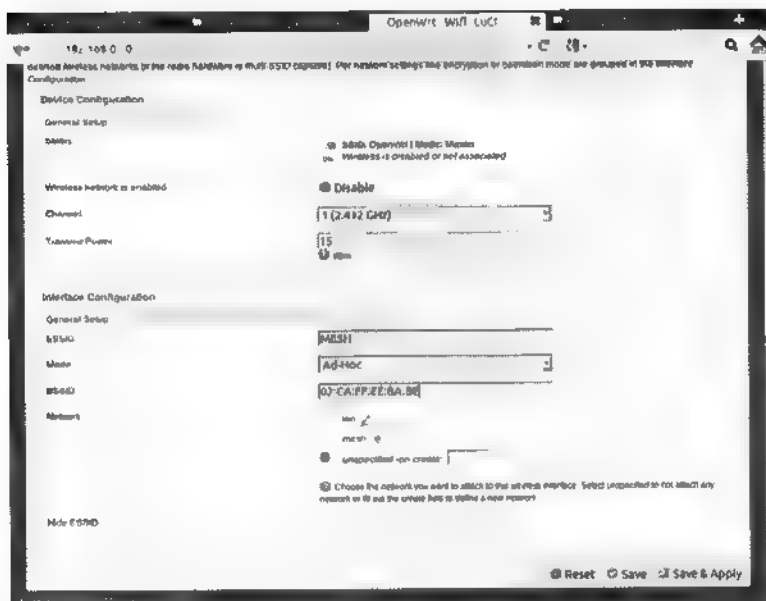
Konfigurasi UBNT, Login ke Web OpenWRT menggunakan password root yang sudah di-set sebelumnya.

Prinsip konfigurasi yang harus dilakukan:

- Set Wireless menjadi ad-hoc
- Set IP Address Interface Wireless
- Konfigurasi olsrd.conf
- Masukan olsrd di rc.local

Terus terang ini semua lebih gampang dilakukan melalui shell. Jadi lakukan

```
ssh root@ip-address-ubnt
```

Set Wireless Interface menjadi ad-hoc:

Melalui Luci WebUI dapat diklik menu:

Network > WiFi > Add

Channel	1 (2.412GHz)
Transmit	Power 15
ESSID	MESH
Mode	Ad-Hoc
BSSID	02:CA:FF:EE:BA:BE
Network	unspecified or create MESH

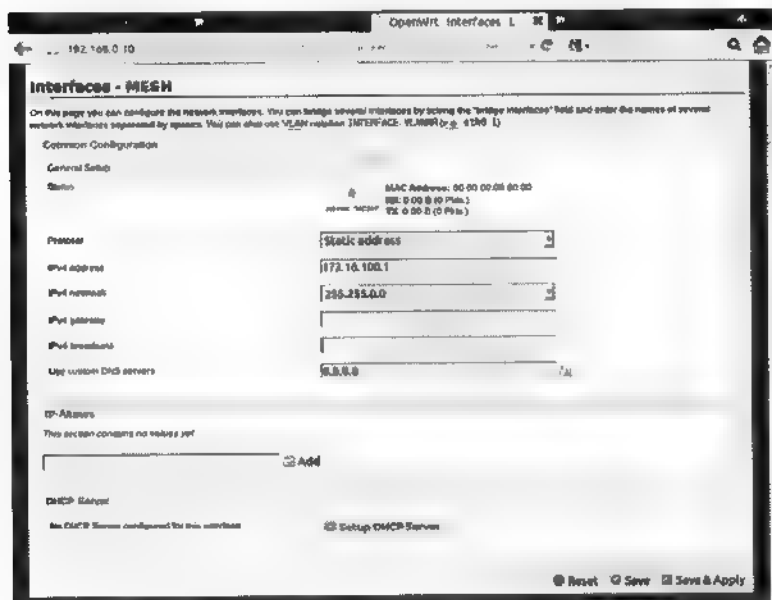
Melalui shell dapat juga diedit:

```
vi /etc/config/wireless
```

Isi dengan:

```
config 'wifi-device' 'wifi0'
    option 'type' 'atheros'
    option 'antenna' 'auto'
    option 'diversity' '0'
    option 'channel' '1'
    option 'disabled' '0'
    option 'txpower' '15'

config 'wifi-iface'
    option 'ssid' 'MESH'
    option 'encryption' 'none'
    option 'device' 'wifi0'
    option 'mode' 'adhoc'
    option 'network' 'mesh'
    option 'bssid' '02:CA:FF:EE:BA:BE'
```

Set IP Address Interface atatik, konfigurasi melalui WebUI:

Melalui WebUI dapat diedit:

Network > Add new interface

Name of the new interface MESH

Protocol of the new interface Static Address

Cover the following interface Wireless Network

SUBMIT

Network > Interfaces MESH > Common Configuration

Protocol Static Address

IPv4 address 172.16.x.x

IPv4 netmask 255.255.0.0

Use custom DNS servers 8.8.8.8

Melalui Shell dapat juga diedit:

```
vi /etc/config/network
```

Isi dengan:

```
config 'interface' 'mesh'
        option '_orig_ifname' 'ath0'
        option '_orig_bridge' 'false'
        option 'proto' 'static'
        option 'ipaddr' '172.16.100.1'
        option 'netmask' '255.255.0.0'
        option 'dns' '8.8.8.8'
```

Konfigurasi olsrd.conf, buat /etc/olsrd.conf:

```
vi /etc/olsrd.conf
```

Isi minimal dengan:

```
Hna4
{
}
Hna6
{
}
LoadPlugin "olsrd_txtinfo.so.0.1"
{
```

```
  PlParam     "accept" "0.0.0.0"
}
InterfaceDefaults
{
}
Interface "ath0" "eth0" "br-lan"
{
  Mode "mesh"
}
```

Masukkan olsrd di rc.local:

Pastikan melalui WebUI, bahwa OLSRD tidak aktif dari OpenWRT Akan lebih stabil menjalankan OLSRD melalui perintah shell dari rc.local.

Melalui WebUI cek:

System > StartUp > olsrd > Disable

Melalui Shell edit:

vi /etc/rc.local

Isi dengan:

/usr/sbin/olsrd -f /etc/olsrd.conf

Monitor OLSR status di Luci UI melalui:

Status > OLSR >

Neighbours

Routes

Topology

HNA

MID

SmartGW

Interfaces

OLSR-Viz

Kita akan menemukan:

- "Neighbour Quality"
- "Neighbour Link Quality"
- "ETX".

Cara membacanya:

NQ parameter 0 (sangat jelek) s/d 1 (sangat baik)

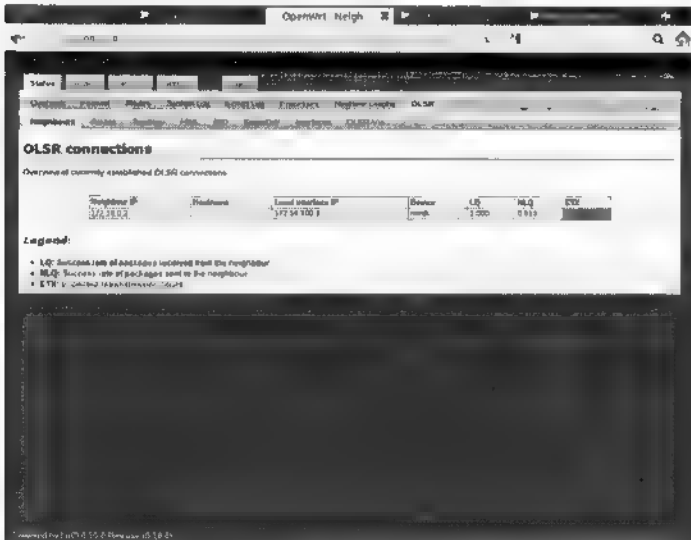
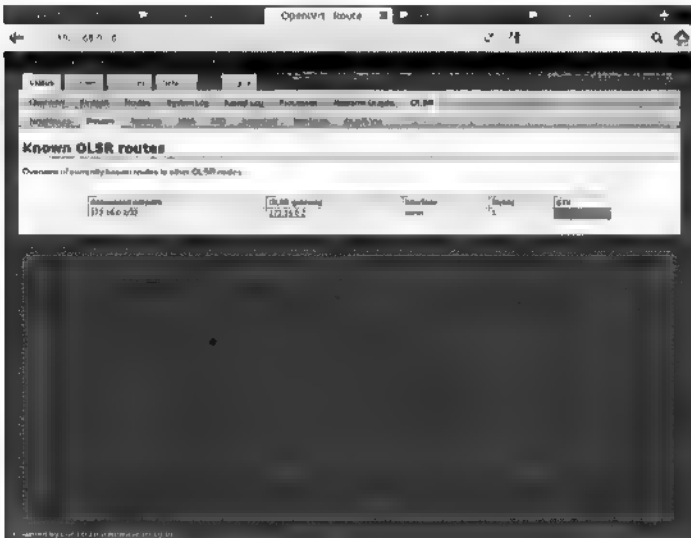
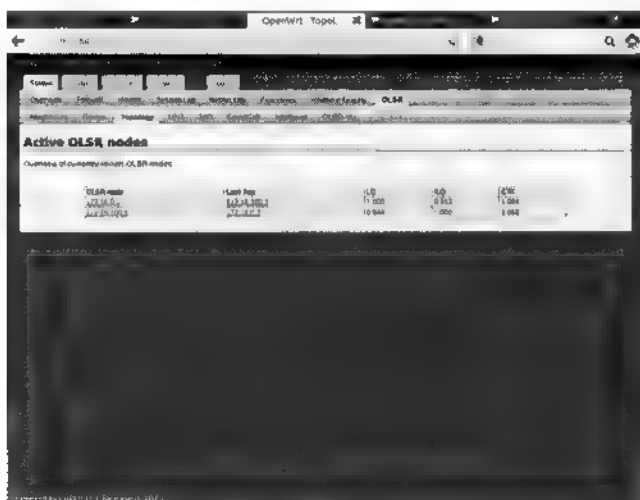
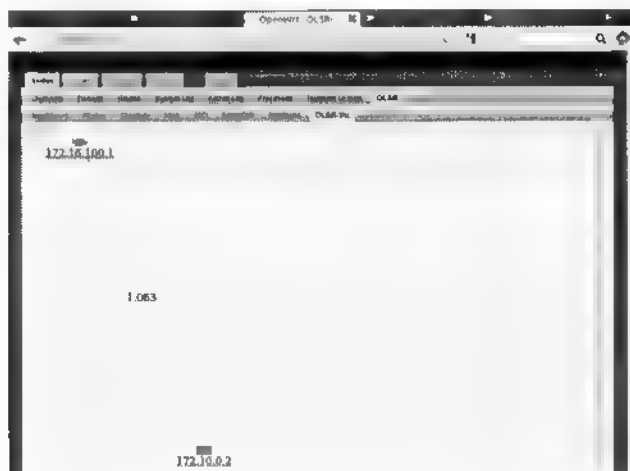
NLQ parameter 0 (sangat jelek) s/d 1 (sangat baik)

ETX paremeter (global transmission quality) dari 1 s/d tak terhingga

ETX antara 1-2 sambungan baik

ETX antara 2-3 sambungan normal

ETX di atas 3 sambungan tidak baik

Tampilan menu Status > OLSR memperlihatkan interface yang menjalankan OLSR, jumlah Neighbours, Nodes, HNA, total link, dll.

Pada menu **Status > OLSR > Neighbours** diperlihatkan tetangga kita yang menjalankan OLSR dan statusnya, terutama perhatikan warna ETX.

Pada menu **Status > OLSR > Routes** kita dapat melihat route ke arah berbagai jaringan melalui OLSR.

Pada menu **Status > OLSR > Topology** kita dapat melihat node OLSR yang aktif di jaringan, maupun kualitas sambungan ke node-node tersebut.

Pada menu Status > OLSR > OlSR-Viz kita dapat melihat visualisasi topologi jaringan mesh yang ada. Hal ini sangat memudahkan kita untuk melihat/membayangkan jaringan yang ada.

KONFIGURASI OLSRD UBNT NANOSTATION2 DENGAN FIRMWARE OPENWRT BUATAN SENDIRI DAN WEBGUI

Konfigurasi UBNT melalui WebGUI tidak berbeda jauh dengan konfigurasi Buffalo dengan WebUI maupun dengan OpenWRT Backfire yang dijelaskan di bagian sebelumnya. Oleh karena itu pembaca disarankan untuk membaca bagian konfigurasi OpenWRT di Buffalo. Memang firmware di Buffalo mempunyai fasilitas lebih banyak karena pada saat mengkompilasi kita menambahkan banyak aplikasi karena memori Buffaloyang lebih besar.

KONFIGURASI OLSRD DI UBNT NANOSTATION2 TANPA WEBGUI

Salah satu contoh kompilasi firmware di bagian sebelumnya adalah firmware OpenWRT yang sengaja dikompilasi tanpa HTTP dan LuCI. Akibatnya, firmware tersebut jadi sangat ringan, karena dioperasikan tanpa WebGUI. Konfigurasinya sebetulnya relatif sederhana . Secara umum langkah yang perlu dilakukan ada empat (4), yaitu:

- Aktifkan Wifi menjadi adhoc network.
- Set IP address agar statik
- Set /etc/olsrd.conf
- Edit /etc/rc.local agar olsrd jalan dengan baik.

Untuk mengaktifkan Wifi kita perlu mengedit:

```
vi /etc/config/wireless
```

Masukkan:

```
config wifi-device  radio0
        option type     mac80211
        option channel  1
        option macaddr  00:15:6d:da:ec:d1
        option hwmode   11g
        option antenna  auto
        option diversity 0
        option txpower  15
        option disabled 0

config wifi-iface
        option device   radio0
        option network  mesh
        option mode     adhoc
        option ssid     MESH
        option encryption none
        option bssid    02:CA:FF:EE:BA:BE
```

Biasanya di konfigurasi awal /etc/config/wireless ada kata-kata:

```
# REMOVE THIS LINE TO ENABLE WIFI:
option disabled 1
```

Ubah menjadi option disable 0.

Edit konfigurasi IP address untuk wireless:

```
vi /etc/config/network
```

Masukkan:

```
config interface mesh
        option ifname   wlan0
        option type     bridge
        option proto    static
        option ipaddr   172.16.123.12
        option netmask  255.255.0.0
```

Restart network:

```
/etc/init.d/network restart
```

Cek di console, apakah sudah operasional dengan baik. Ketik perintah:

```
iwconfig
```

akan keluar kurang lebih:

```
wlan0     IEEE 802.11bg  ESSID:"MESH"
          Mode:Ad-Hoc    Frequency:2.412    GHz    Cell:
02:CA:FF:EE:BA:BE
          Tx-Power=15 dBm
          RTS thr:off   Fragment thr:off
          Encryption key:off
          Power Management:on
```

Edit konfigurasi olsrd:

```
vi /etc/olsrd.conf
```

Masukkan:

```
Hna4
{
}
Hna6
{
}
LoadPlugin "olsrd_txtinfo.so.0.1"
{
   PlParam     "accept" "0.0.0.0"
}
InterfaceDefaults
{
}
Interface "wlan0" "eth0" "br-lan"
{
   Mode "mesh"
}
```

Masukkan ke /etc/rc.local

```
ifconfig wlan0 172.16.123.12
/etc/init.d/olsrd stop
```

```
rm /var/run/olsrd-ipv4.lock
olsrd -f /etc/olsrd.conf &
```

Selesai sudah proses konfigurasi UBNT NanoStation2. Agar menjadi Mesh Node, kita perlu me-restart menggunakan perintah:

```
reboot
```

INSTALASI APLIKASI PENDUKUNG OLSRD DI BUFFALO WZRHPG450H MENGGUNAKAN OPENWRT

Bagi Anda yang menggunakan Buffalo WZR HP G450H dengan firmware OpenWRT dengan WebGUI, berikut adalah contoh instalasi aplikasi pendukung OLSRD di Buffalo tersebut.

Setelah Buffalo di-flash menggunakan OpenWRT, pada laptop lakukan:

```
dhclient eth0
```

ping ke 192.168.1.1, harusnya dapat respons:

```
ping 192.168.1.1
```

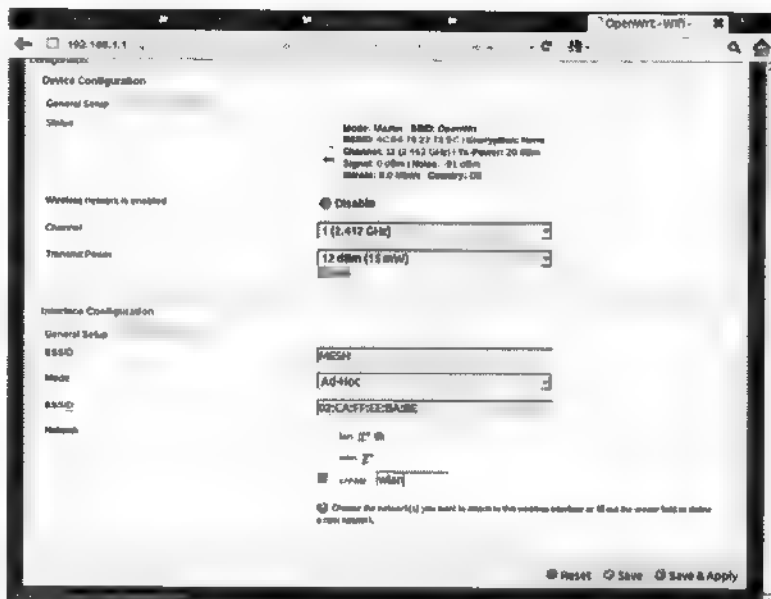
Telnet ke Buffalo dari laptop:

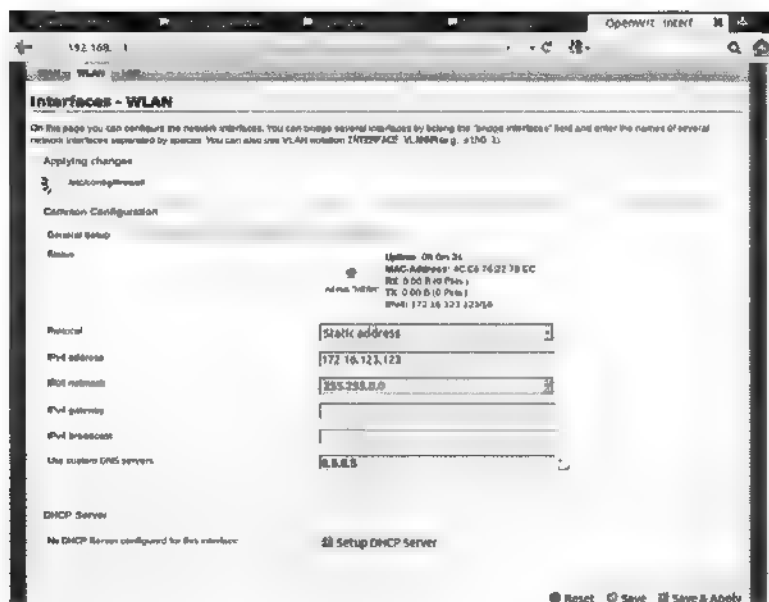
```
telnet 192.168.1.1
```

Ubah password root OpenWRT:

```
passwd
masukan password root)
(masukan password root)
exit
```

Aktifkan MESH.

Pertama-tama aktifkan dulu interface WLAN. Akses melalui Web, masuk ke menu **Administration > Network > Interfaces > Wifi > Device Configuration.**

Channel	1 (2.412 GHz)
Transmit Power	20dBm (100mW)
ESSID	MESH
Mode	ad-hoc
BSSID	02:CA:FF:EE:BA:BE
Network	
create:	wlan

Selanjutnya kita dapat menset interface WLAN melalui menu Administration > Interfaces > WLAN.

Melalui General Setup > Protocol > Static Address, set supaya:

IPv4 address

IPv4 netmask

IPv4 gateway

IPv4 broadcast

Kita set dengan benar. Alternatif lain, kita dapat melakukan proses melalui terminal. Ssh ke Buffalo dari laptop:

```
ssh root@192.168.1.1
```

Edit:

```
vi /etc/config/network
```

Isi dengan:

```
config interface 'wlan'
    option _orig_ifname 'wlan0'
    option _orig_bridge 'false'
    option proto 'static'
    option ipaddr '172.16.123.123'
    option netmask '255.255.0.0'
    option dns '8.8.8.8'
```

Cek melalui shell:

```
iwconfig
```

Harus keluar kira-kira:

```
wlan0     IEEE 802.11bgn  ESSID:"MESH"
Mode:Ad-Hoc  Frequency:2.412 GHz  Cell: 02:CA:FF:EE:BA:BE
Tx-Power=12 dBm
RTS thr:off   Fragment thr:off
Encryption key:off
Power Management:on
```

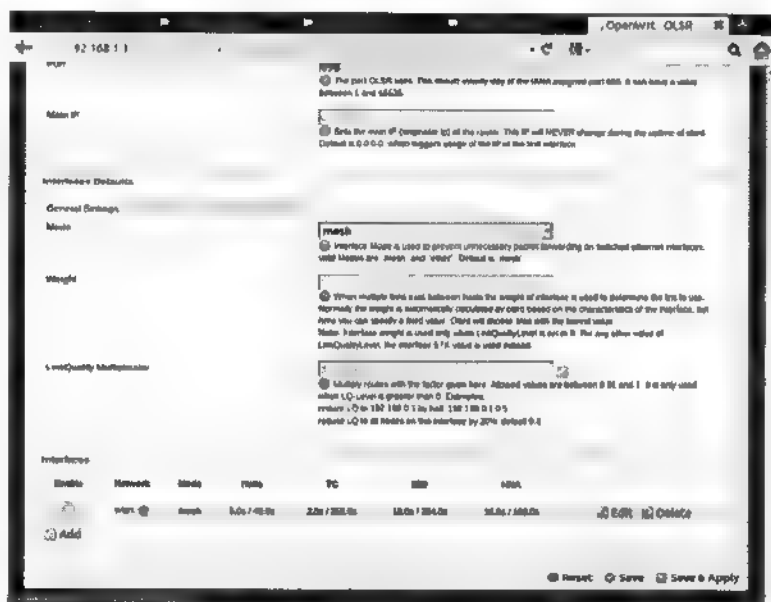
Konfigurasi olsrd.conf, buat /etc/olsrd.conf:

```
vi /etc/olsrd.conf
```

Isi minimal dengan:

```
hna4
{
}
```

```
Hna6
{
}
LoadPlugin "olsrd_txtinfo.so.0.1"
{
   PlParam     "accept" "0.0.0.0"
}
InterfaceDefaults
{
}
Interface "wlan0" "eth0" "br-lan"
{
   Mode "mesh"
}
```

Pastikan melalui WebUI, bahwa OLSRD tidak aktif dari OpenWRT. Masuk ke menu Administration > OLSR > Interfaces > Dibuang ENABLE.

Akan lebih stabil menjalankan OLSRD melalui perintah shell dari rc.local. Restart aplikasi

```
olsrd -f /etc/olsrd.conf
```

Sebaiknya masukkan juga perintah di atas ke:

```
vi /etc/rc.local
```

Tambahkan:

```
olsrd -f /etc/olsrd.conf &
```

agar kalau Buffalo di-boot OLSR akan jalan sendiri.

SEDIKIT CATATAN TENTANG FIREWALL DI OPENWRT

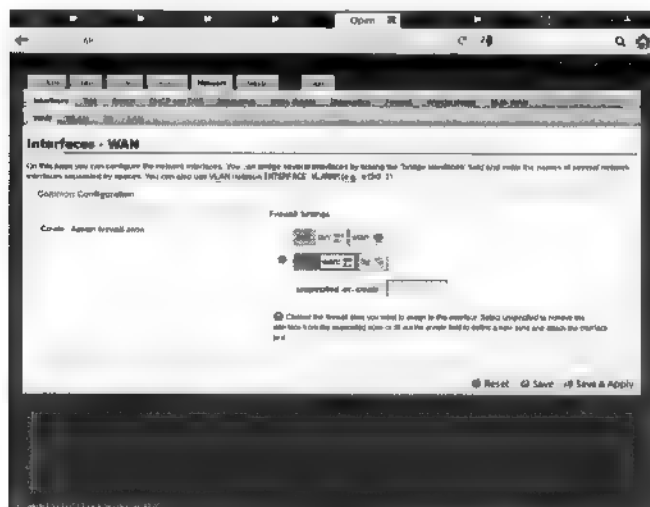
Firewall di OpenWRT sebetulnya relatif sederhana dan mudah digunakan. Secara umum kita hanya butuh dua (2) zone, yaitu

- LAN (tidak ada proxy & relatif terbuka)
- WAN (ada proxy)

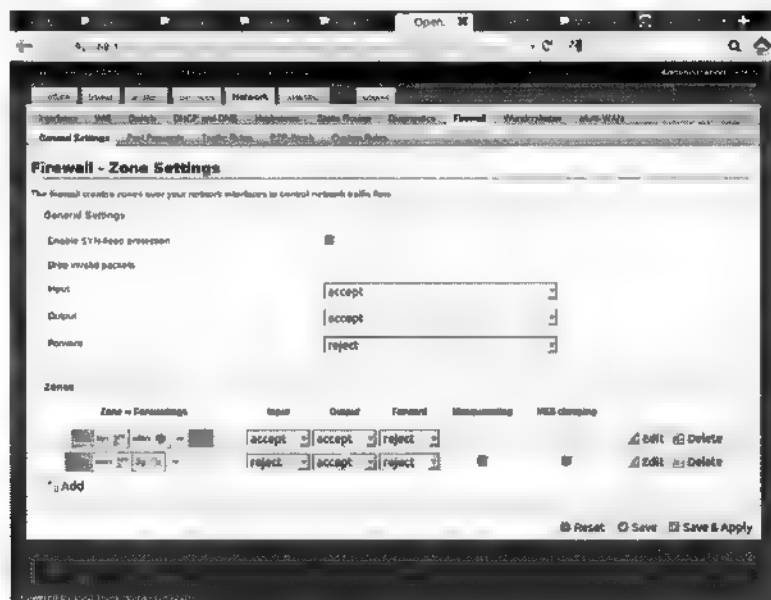
Semua traffic ke WAN akan di-proxy.

Jika kita membuka/mengaktifkan sebuah interface baru, kita perlu men-set firewall. Secara umum ada dua (2) tahap, yaitu:

- Menset interface termasuk pada zone firewall yang mana.
- Mengatur zone.

Untuk mengatur interface masuk pada sebuah zone firewall yang mana, kita dapat masuk ke menu **Administration > Network > Interfaces > (nama interface) > Firewall.** Sebaiknya kita klik salah satu, LAN atau WAN firewall, untuk interface tersebut.

Setelah semua interface di-set masuk ke zone firewall, langkah selanjutnya adalah men-set Zone Firewall, terutama mengatur paket yang diijinkan lewat, misalnya:

WAN	tidak menerima paket masuk, hanya mengijinkan paket keluar saja.
WAN	melakukan masquerade (proxy)
LAN	dapat mengirim paket ke WAN

Menu untuk mengatur ini semua sangat sederhana dan dapat diakses melalui menu **Administration > Network > Firewall > General Settings.**

SAMBUNGAN KE INTERNET DAN JARINGAN LAIN

Pada bagian ini akan dijelaskan beberapa cara untuk menyambungkan jaringan wireless MESH ke Internet maupun jaringan lain. Secara umum ada beberapa skenario, yaitu:

- Sambungan menggunakan LAN ke Internet
- Sambungan menggunakan LAN ke jaringan lain / LAN.
- Sambungan menggunakan modem 3G.
- Banyak sambungan ke Internet.

SAMBUNGAN MENGGUNAKAN LAN KE INTERNET

Jika kita mempunyai sambungan ke Internet menggunakan LAN, biasanya menggunakan Speedy dan router Buffalo diletakkan di belakang modem Speedy. Yang perlu kita lakukan di OLSR tidak banyak, hanya mengedit file:

```
vi /etc/olsrd.conf
```

Isi pada entry:

```
Hna4
{
        0.0.0.0/0
}
```

Ini akan memberitahukan ke semua node OLSR yang beroperasi bahwa komputer kita bersedia untuk menjadi sambungan ke Internet bagi yang lain.

SAMBUNGAN MENGGUNAKAN LAN KE JARINGAN LAIN/LAN

Bila kita mempunyai sambungan LAN di belakang node OLSR yang kita operasikan, kita perlu memberitahu semua node OLSR yang lain akan keberadaan network LAN tersebut.

Misalnya LAN di belakang kita adalah:

```
192.168.123.0/24
```

Maka kita perlu mengedit file:

```
vi /etc/olsrd.conf
```

Isi pada entry:

```
Hna4
{
        192.168.123.0/24
}
```

SAMBUNGAN MENGGUNAKAN MODEM 3G

Saat build firmware, tambahkan paket:

```
./scripts/feeds update packages
./scripts/feeds install usb-modeswitch usb-modeswitch-data usbutils multiwan
```

Pastikan saat konfigurasi "make menuconfig", parameter berikut di-set dengan benar,

Network >

chat

ppp

ppp-mod-pppoa

ppp-mod-pppoe

ppp-mod-pppol2tp

ppp-mod-pptp

ppp-mod-radius

ppp-multilink

pppdump

pppstats

Kernel Modules > USB Support >

kmod-usb-serial

kmod-usb-serial-ipw

kmod-usb-serial-option

kmod-usb-serial-qualcomm

kmod-usb-serial-wwan

kmod-usb-acm

LuCI >

6. Protocols >

luci-proto-3g

Instalasi USB Modem. Masukkan USB modem & Restart Router. Cek dmesg melalui shell ketik

```
dmesg
```

akan ada kata-kata seperti:

```
[   18.140000] USB Serial support registered for GSM modem (1-port)
[  18.150000] usbcore: registered new interface driver option
[  18.150000] option: v0.7.2:USB Driver for GSM modems
[   18.270000] USB Serial support registered for Qualcomm USB modem
[     18.280000] usbcore: registered new interface driver qcserial
[   24.540000] usb 1-1: new high-speed USB device number 3 using ehci-platform
[   24.710000] option 1-1:1.0: GSM modem (1-port) converter detected
[   24.710000] usb 1-1: GSM modem (1-port) converter now attached to ttyUSB0
[   24.720000] option 1-1:1.1: GSM modem (1-port) converter detected
[   24.730000] usb 1-1: GSM modem (1 port) converter now attached to ttyUSB1
[   25.750000] scsi 3:0:0:0: CD-ROM            HUAWEI   Mass Storage    2.31 PQ: 0 ANSI: 2
[   25.760000] scsi 4:0:0:0: Direct-Access     HUAWEI   MMC Storage     2.31 PQ: 0 ANSI: 2
```

```
[   18.140000] USB Serial support registered for GSM modem (1-port)
[  18.150000] usbcore: registered new interface driver option
[  18.150000] option: v0.7.2:USB Driver for GSM modems
[   18.270000] USB Serial support registered for Qualcomm USB modem
[    18.280000] usbcore: registered new interface driver qcserial
[  24.540000] usb 1-1: new high-speed USB device number 3 using ehci-platform
[  24.710000] option 1-1:1.0: GSM modem (1-port) converter detected
[   24.710000] usb 1-1: GSM modem (1-port) converter now attached to ttyUSB0
[  24.720000] option 1-1:1.1: GSM modem (1-port) converter detected
[   24.730000] usb 1-1: GSM modem (1-port) converter now attached to ttyUSB1
```

Kita juga dapat melakukan:

```
cat /proc/bus/usb/devices
```

Keluar kata-kata kurang lebih:

```
T:  Bus=01 Lev=01 Prnt=01 Port=00 Cnt=01 Dev#=  3 Spd=480 MxCh= 0
D:  Ver= 2.00 Cls=00(>ifc ) Sub=00 Prot=00 MxPS=64 #Cfgs=  1
P:  Vendor=12d1 ProdID=1003 Rev= 0.00
```

```
S: Manufacturer=HUAWEI Technology
S: Product=HUAWEI Mobile
```

Cek /dev/ttyUSB*

```
ls /dev/ttyUSB*
```

Akan keluar misalnya:

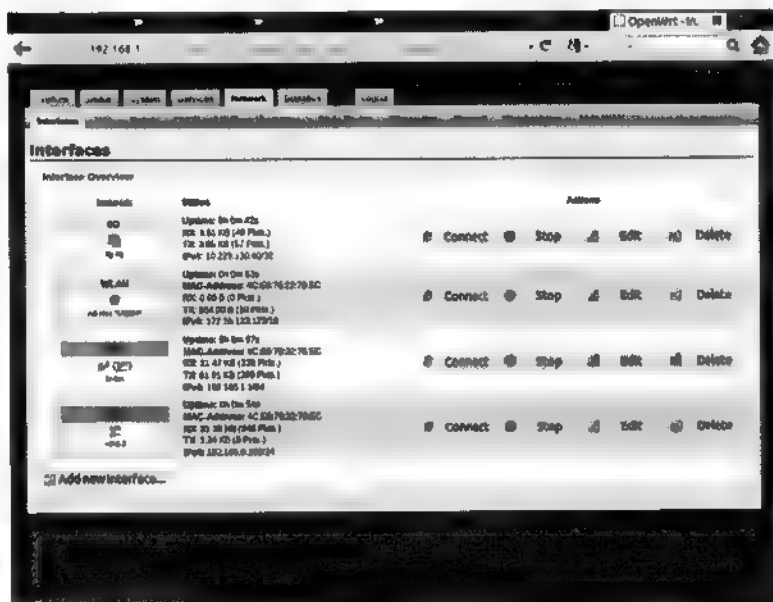
```
/dev/ttyUSB0  /dev/ttyUSB1
```

Salah satu di antaranya adalah USB modem yang kita pasang. Selanjutnya, konfigurasi USB Modem. Edit file

```
vi /etc/config/network
```

Tambahkan:

```
config interface 3g
  option ifname ppp0
  # option pincode 1234
  option device /dev/ttyUSB0
  # option apn your.apn
  option apn internet
  option service umts
  option proto 3g
```

Selanjutnya kita dapat me-reboot router. Biasanya akan secara otomatis modem 3G terdeteksi dan akan langsung bisa diaktifkan/digunakan. Kita dapat melihat statusnya di Administration > Network > Interfaces.

Jika kita ingin mematikan sambungan 3G, melalui menu **Administration > Network > Interfaces, tekan Stop.**

BANYAK SAMBUNGAN KE INTERNET

Bagian ini merupakan skenario yang paling seru. Bayangkan sebuah router mempunyai beberapa sambungan ke Internet. Kita ingin menggabungkan bandwidth yang ada agar total bandwidth menjadi jumlah dari masing-masing bandwidth sambungan.

Yang perlu disiapkan

- Router Buffalo
- Install OpenWRT buatan sendiri, compile multiwan di dalamnya.

Pastikan parameter pada "make menuconfig" berikut di-set dengan benar:

Network >

multiwan

LuCI >

3. Applications >

luci-app-multiwan

Untuk konfigurasi MultiWAN, kita cukup masuk ke menu Administration > Network > MultiWAN. Sebetulnya tidak banyak yang perlu di-set, yaitu:

Enable di contreng

Masukan nama interface WAN > Add

Masukan nama interface 3G > Add

Default route (isi misalnya wan).

Konfigurasi interface tidak banyak, terutama:

Load Balancer Distribution (untuk mengatur beban pada interface, beri beban kecil pada 3G).

Fail over traffic destination (untuk mengatur kalau ada kegagalan).

PENUTUP

Pada hari ini, khususnya di kalangan anak muda, Internet menjadi kebutuhan mendasar yang sulit untuk ditinggalkan. Sayang sekali, khususnya di daerah yang jauh dari ibu kota propinsi atau ibu kota kabupaten/kota, Internet, bahkan jaringan komputer, masih menjadi barang sangat mewah.

Buku ini memberikan solusi sederhana, yang mana dengan menggunakan komputer apa adanya kita dapat dapat membangun jaringan dengan menyelipkan kemampuan untuk me-relay paket antarkomputer yang ada di lokasi yang sama. Teknik gotong royong yang sangat sederhana ini kemudian dikenal dengan nama jaringan mesh.

Penggunaan teknik mesh memungkinkan kita untuk membangun jaringan yang sangat murah menggunakan komputer yang kita gunakan. Teknik ini jelas sangat mudah dibangun di kampus-kampus. Teknik ini juga sangat berguna bagi kita yang membangun di daerah pegunungan yang sangat sulit untuk membangun relay yang banyak. Jaringan ini terutama sangat membantu bagi daerah kepulauan seperti Maluku, Ambon, Ternate, Nusa Tenggara, juga pedalaman Irian.

Mudah-mudahan buku ini dapat memberikan inspirasi bagi mereka yang ingin membangun jaringan yang mudah di pedalaman Indonesia.

REFERENSI

- http://opensource.telkomspeedy.com/wiki/index.php/Mesh
- http://opensource.telkomspeedy.com/wiki/index.php/OLSR
- http://wiki.freifunk.net/Kategorie:English
- https://forum.openwrt.org/viewtopic.php?id=15201
- http://wiki.openwrt.org/doc/howto/build
- http://downloads.openwrt.org/docs/buildroot-documentation.html

TENTANG PENULIS

Onno Widodo Purbo adalah seorang tokoh teknologi informasi Indonesia. Ayah Onno, Hasan Poerbo, adalah profesor di ITB bidang lingkungan hidup yang banyak memihak rakyat kecil. Onno masuk ITB pada jurusan Teknik Elektro angkatan 1981. Enam tahun kemudian Onno lulus dengan predikat wisudawan terbaik. Setelah itu Onno melanjutkan studi ke Kanada dengan beasiswa dari PAU-ME. RT/RW-Net adalah salah satu dari sekian banyak gagasan yang Onno lontarkan yang mengukir sejarah Internet Indonesia. Ia juga aktif menulis dalam bidang teknologi informasi di berbagai media, seminar, konferensi nasional maupun internasional, terutama untuk memberdayakan masyarakat Indonesia menuju masyarakat berbasis pengetahuan. Lebih dari 161 kota besar di 33 negara ia kunjungi dalam perjalanan hidupnya. Onno percaya filosofi copyleft. Banyak tulisannya dipublikasikan secara gratis di internet. Saat ini aktif membantu Prof. Yohanes Surya di STKIP Surya untuk menelurkan guru-guru di Indonesia agar lebih melek komputer. Semoga dapat memberikan perubahan yang besar bagi bangsa Indonesia.